91-30

Recognizing and Incrementally Evolving Texture Concepts in Dynamic Environments: An Incremental Model Generalization Approach

Peter W. Pachowicz

P91-~~28~~ 30
MLI 91-10

Report Documentation Page		*Form Approved* *OMB No. 0704-0188*
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.		
1. REPORT DATE **NOV 1991**	2. REPORT TYPE	3. DATES COVERED **00-00-1991 to 00-00-1991**
4. TITLE AND SUBTITLE **Recognizing and Incrementally Evolving Texture Concepts in Dynamic Environments: A model generalization approach**		5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **George Mason University,Center for Artificial Intelligence,Fairfax,VA,22030**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S) 11. SPONSOR/MONITOR'S REPORT NUMBER(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release; distribution unlimited**		
13. SUPPLEMENTARY NOTES		
14. ABSTRACT **The paper presents a novel approach to the invariant recognition of objects (through texture) in dynamic environments. The proposed approach assumes that (1) the system has to recognize objects on each image of a sequence of images, (2) the images demonstrate the variability ofconditions under which objects are perceived (resolution, lighting, positioning), (3) both an observer and the objects can move, (4) the extraction of texture attributes and training examples can be imperfect. and (5) the system has to work autonomously (i.e., without the aid of a teacher). We propose to utilize images of a sequence to adapt system models to perceived variabilities of texture characteristics. Such an adaptation integrates recognition and segmentation processes of computer vision with incremental knowledge acquisition processes of machine learning. While the initial acquisition of texture models is driven by a teacher, the evolution of these models is performed over a sequence of images without the help of a teacher. Initially acquired texture descriptions are applied to recognize and extract objects on the next images. The effectiveness of such recognition and object extraction is monitored. When this effectiveness decreases, the system selects new training data and activates learning processes to improve its models. This paper presents both an outline of the iterative evolution methodology and the investigation of an incremental model generalization approach as a part of the evolution of texture models. Experiments were run in a partially-supervised mode rather than a fully autonomous model evolution. The experiments are compared based on the following three system configurations: (i) a one-level control structure, (ii) a two-level control structure, and (iii) a two-level control structure with data filtering. Obtained results are evaluated using the criteria of system recognition effectiveness, recognition stability, and predictability of evolved models.**		
15. SUBJECT TERMS		

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**54**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

RECOGNIZING AND INCREMENTALLY EVOLVING TEXTURE CONCEPTS IN DYNAMIC ENVIRONMENTS:
A model generalization approach

Peter W. Pachowicz

Center for Artificial Intelligence
George Mason University
Fairfax, VA 22030

Key words: computer vision through machine learning, intelligent autonomous systems in dynamic environments, incremental model evolution.

ABSTRACT

The paper presents a novel approach to the invariant recognition of objects (through texture) in dynamic environments. The proposed approach assumes that (1) the system has to recognize objects on each image of a sequence of images, (2) the images demonstrate the variability of conditions under which objects are perceived (resolution, lighting, positioning), (3) both an observer and the objects can move, (4) the extraction of texture attributes and training examples can be imperfect, and (5) the system has to work autonomously (i.e., without the aid of a teacher). We propose to utilize images of a sequence to adapt system models to perceived variabilities of texture characteristics. Such an adaptation integrates recognition and segmentation processes of computer vision with incremental knowledge acquisition processes of machine learning. While the initial acquisition of texture models is driven by a teacher, the evolution of these models is performed over a sequence of images without the help of a teacher. Initially acquired texture descriptions are applied to recognize and extract objects on the next images. The effectiveness of such recognition and object extraction is monitored. When this effectiveness decreases, the system selects new training data and activates learning processes to improve its models.

This paper presents both an outline of the iterative evolution methodology and the investigation of an incremental model generalization approach as a part of the evolution of texture models. Experiments were run in a partially-supervised mode rather than a fully autonomous model evolution. The experiments are compared based on the following three system configurations: (i) a one-level control structure, (ii) a two-level control structure, and (iii) a two-level control structure with data filtering. Obtained results are evaluated using the criteria of system recognition effectiveness, recognition stability, and predictability of evolved models.

TABLE of CONTENTS

1. INTRODUCTION AND MOTIVATION

1.1. Introduction

Interdisciplinary research conducted in intelligent systems has became one of the most challenging and fascinating areas in the field of computer science and engineering. Traditional control engineering tools (Jamshidi, 1983) and pattern recognition tools (Nillson, 1965, Duda and Hart, 1973) broadly used for several decades still cannot deal with the growing complexity of engineering autonomous systems. The requirements for these systems, determined by the application domains, are still growing. The increasing complexity of such integrated systems and their ever growing engineering requirements, however, push towards the development and application of very sophisticated artificial intelligence (AI) methodologies never explored before in the area of autonomous intelligent systems. Machine learning is such an emerging technology that is applied and investigated to assist us in solving problems of computer vision presented in this paper.

Generally speaking, researchers have achieved significant progress in isolated areas such as image understanding, decision making, planning, speech recognition, expert systems, model and knowledge representation, machine learning, etc. Tools in these separated areas, however, must be integrated when one considers developing an autonomous system working within a real-world environment. This integration includes the combination of different intelligent behaviors. Researchers have advanced such integration through the development of architectures for intelligent autonomous systems, for example, blackboard architecture (Nii, 1986), schema architecture (Draper, et al., 1989, Arkin, 1990), pipeline architecture (Goto, et al., 1988), behavioral architecture (Brooks, 1990, Arkin, 1990), waterfall architecture (Spiessbach, 1989). Some of these architectures have already been integrated within a single autonomous platform, for example, the NavLab autonomous vehicle (Goto and Stentz, 1987).

One of the primary goals in the development of early autonomous systems was to build and then to integrate conceptually different system modules to perform a given task. A task was for the most part defined with respect to technical rather than environmental realities. This means that the task was executed under strong assumptions that limit system complexity; for example, a static environment, known terrain, known or simplified objects, extended background knowledge, etc. Results evaluation was then focused on the system capability to perform a given task successfully. The success of developed methodologies and systems was demonstrated through experiments. In the same time, researchers analyzed potential issues for future progress. These issues are easily distinguished when one weakens the previously stated strong assumptions; for example, task complexity, environment complexity, object classes, variability of conditions, classes of unexpected situations.

1.2. Characteristics of Dynamic Environments

If a system is created using an assumption that the environment is static then the system autonomy that deals with time variability of environmental characteristics is highly limited. Most outdoor environments, however, are dynamic in their nature. Objects in these environments can move. Moreover, the conditions under which objects are perceived or the system reacts on objects can change. If such conditions vary over time then an autonomous system must adapt to these changes. Such an adaptation is crucial for the robust execution of a given mission.

Considering system perception in outdoor environments, most vision research has been focused on recognizing textures under stationary conditions, i.e., for stable lighting conditions, resolution and surface positioning (Rosenfeld and Davis, 1979). Relatively little has been done on the problem of recognizing textures under dynamic conditions. The problem of recognizing textures under non-stationary conditions occurs in most situations and is therefore of significant practical importance. For example, let us consider the following scenario:

an observer (e.g., an autonomous land vehicle) is moving through an outdoor environment recognizing unstructured objects.

The primary aspect of this scenario is that a system has to recognize objects on images acquired over time. Images of such a sequence are affected by the variability of conditions under which objects are perceived. In order to recognize an object on the forthcoming image of a sequence, the system has to update its texture models regarding changes in the characteristics of visual objects perceived through previous images.

Circumstances that cause the variability of external conditions can be divided into at least the following two groups: (1) projection variability (e.g., changing resolution, different surface orientation) and (2) the influence of natural agents (e.g., changing illumination).

Changing resolution causes both the repetition rate of texture patterns and the characteristics of a single pattern to vary. The variability of the repetition rate is primarily used (as a positive effect) to determine surface orientation in the 3-D space (Kanatani and Chou, 1989). However, the influence of changing resolution on pattern characteristics is highly negative and difficult to predict when classifying a pattern to a certain class of texture. The increase in resolution brings new primitives into evidence, while the decrease in resolution blurs texture towards a constant tone, and some pattern elements may be no longer visible. The variability of texture features obtained for different resolutions has been demonstrated by Roan, et al. (1987), and by Unser and Eden (1989). In several cases, the variability

of texture features can be approximated and used to predict features for the other resolutions. The extrapolation of such characteristics, however, can be totally inefficient.

Changing surface orientation influences the projection of 3-D microstructure of a surface onto a 2-D image. Natural texture perceived from different orientations to the texture surface can show new texture elements while hiding other elements by partial or full occlusion. Moreover, an angled projection of a surface introduces variability in its resolution.

Changing lighting is caused by the characteristics of the light source (e.g., intensity, light spectrum), its position in relation to the object surface, and irregularities such as shadows or highlights. For example, the variability of the texture model caused by changing lighting can easily be observed for specific textures composed of large 3-D material structures. These structures can reflect propagated light differently. In particular, sharp shadows can create a light subtexture overlapped with physical texture.

1.3. Proposed Approach to Texture Recognition

As indicated by Rosenfeld and his panelists (1986), relatively little research effort has been conducted on the development of new methodologies useful for high-level vision, when compared with an immense amount of research conducted on low-level vision. Particularly little has been done in the area of applying machine learning methods to the adaptability of vision systems. The problems of applying symbolic machine learning to texture recognition tasks have barely been explored. Most works on adaptive vision systems are limited, for example, to the improvement of unsupervised image segmentation (e.g., Hsiao and Sawchuk, 1989, Bhanu, et al., 1989, 1990), and the application of expert systems to the automatic configuration of vision systems (Liedtke and Ender, 1986, Neumann, 1987, Matsuyama, 1989, Niemann, et al., 1990). Relatively less effort is paid to the adaptability of object recognition systems in a dynamic environment.

Researchers generally try to apply more powerful methods to the acquisition of texture characteristics and then apply acquired characteristics to the recognition of unknown texture samples; e.g., Markov random fields (Cross and Jain, 1983), Gibbs random fields (Derin and Elliott, 1987), and the spatial frequency spectrum (Weszka, et a., 1976, Liu and Jernigan, 1990). Such a recognition approach belongs to the class of feature extraction oriented methods, where extracting relevant features plays a very important role during training and recognition phases (DuBuf, et al., 1990). The main problem with such a traditional texture recognition approach is that we do not have a universal feature extraction method that works effectively with noisy and imperfect data. When one considers that training data can be noisy and imperfect, one has to agree that the derived descriptions of texture classes can also

contain noisy components. The development of highly specialized feature extraction methods is also questionable because these features have to be sensitive within a wider range of texture occurrences.

In this paper, we present a new approach to the texture recognition problem that deals with the dynamics of the environment, i.e., the variability of external conditions under which textures are perceived. The new approach integrates learning and recognition processes within a closed loop to update texture models. Analysis of system recognition effectiveness, performed over a sequence of images, detects changes in texture occurrences. If this effectiveness, based on the changes, decreases then the system activates incremental learning processes of model evolution to improve the model discriminating power. The system learns initial texture descriptions from teacher-provided examples. Then, thereafter, the system is able to update these descriptions automatically without teacher help.

Variability of texture requires the development of system capabilities that will reconfigure, tune, and update models (knowledge) with respect to observer/object movement and the variation of sensing conditions (changing resolution, lighting, surface positioning, and overlapping noise). An approach presented in this paper investigates the adaptability of vision systems through the development and implementation of an incremental model evolution methodology (Pachowicz, 1991). Specific research topics explored in this paper include:

(1) The application of a symbolic learning methodology, working in an incremental mode, to the evolution of texture descriptions,
(2) The integration of recognition and learning processes within a closed-loop system working without teacher help, and
(3) Empirical investigation of a "partially supervised" model evolution through incremental generalization of texture models over new training examples.

The application of machine learning methodology to the acquisition of texture descriptions gives us an opportunity to manipulate acquired models (knowledge) in order to improve them and match them to testing data. A symbolic machine learning approach also gives an opportunity to acquire hybrid models, i.e., models that include numeric, symbolic and structural attributes. Recently, we illustrated the advantages of such an approach to a texture recognition problem (Pachowicz, 1990, Bala and Pachowicz, 1990, Pachowicz and Bala, 1991).

2. AN OUTLINE OF THE METHODOLOGY FOR AUTONOMOUS MODEL EVOLUTION

This section presents an overview of the methodology developed to incrementally evolve texture models in order to recognize objects within a dynamic environment. We introduce an integrated

system architecture, investigate control structures used to evolve models, an evolution cycle, and various factors of learning-based model evolution technology.

Related research work has been reported by Goldfarb (1990). Since model evolution integrates multi-disciplinary research, Goldfarb introduces "Pattern Learning" to integrate symbol formation and recognition processes with Artificial Intelligence symbol manipulation processes. He understands system evolution as system adaptation through learning from the environment, where a system's *"structure"* is modified dynamically during such learning. He focuses on the design of a learning system called a transformation system which is both a learning and a recognition system. The recognition of structural patterns and a neural net are proposed for its implementation. Since no experiments with real objects are presented, the dynamic performance of his system (i.e., continuous interactions with dynamic environment) is still questionable under the assumption that the system has to perceive and represent variable object characteristics.

2.1. Integrated System Architecture

Since the perceived texture characteristics of objects can change in a dynamic environment, an adaptive texture recognition system has to both detect these changes and to update its texture models. Such a task must integrate computer vision (CV) with concept acquisition and evolution (CA&E). An interesting feature of such an integration is that both methods have to work in a dynamic regime; i.e., over a sequence of images. Moreover, cooperation of both methods can be arranged by a teacher only at the beginning. Later they have to work together over an incoming sequence of images, autonomously.

The global architecture of such an integrated system is presented in Figure 1. The architecture shows the highly modular structure of the entire system. The designing principle was to distribute vision and concept acquisition/evolution processes for their implementation on parallel hardware allowing for modifications.

The input to the vision system is a sequence of images $P(t)$ obtained over time. Images of such a sequence represent variable perceptual conditions (resolution, lighting, object positioning) caused by the movement of an autonomous system and the observed objects, and/or the dynamics of natural agents. These images are obtained in a certain time interval ΔT. The time interval, however, can be constant or it can be adaptive on-line, depending upon the velocity of changes within environment and the speed of a robot vehicle. Directly measured sensor parameters characterizing overall external conditions can be transformed to the CA&E system, as can sensor settings.

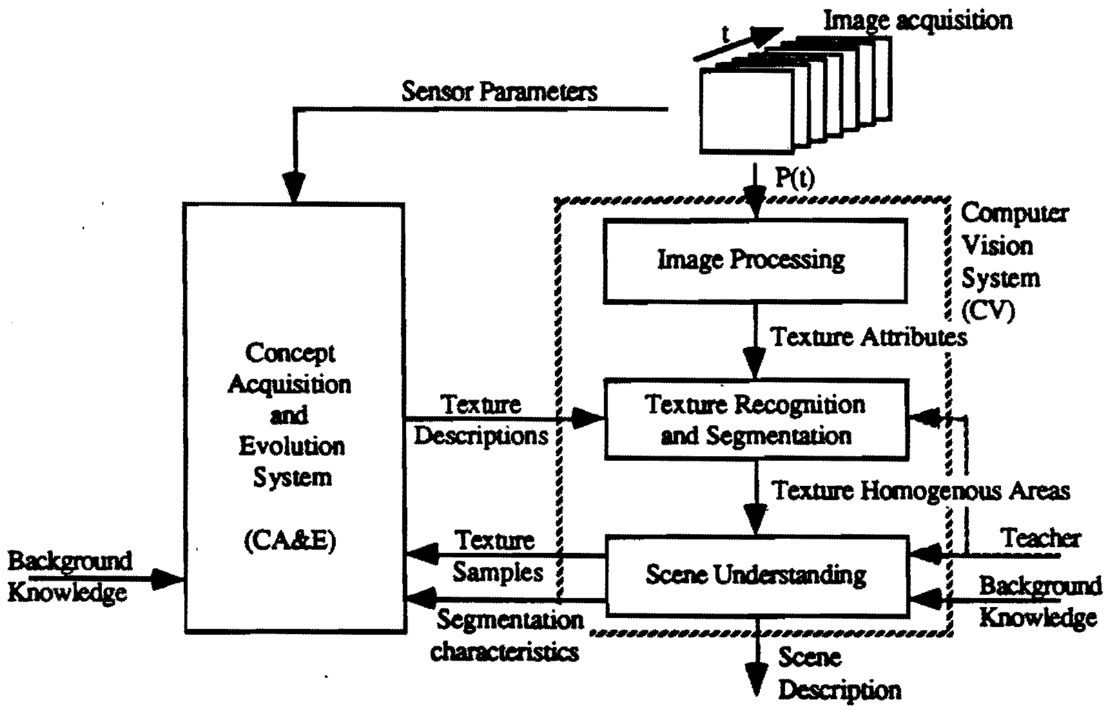

Fig.1 Integrating Computer Vision (CV) with Concept Acquisition and Evolution (CA&E) for texture recognition in dynamic environments

The CV system performs the following three sequential tasks: (i) image processing, (ii) texture recognition and segmentation, and (iii) scene understanding. These tasks are performed on a single image provided by a sensor. The first task, image processing, processes an incoming image to extract texture attributes. The second task, texture recognition and segmentation, applies texture concept descriptions to recognize extracted vectors of attributes. Then, the segmentation process unifies homogeneous texture areas of the same class. The third task, scene understanding, formulates a scene description and evaluates recognition and segmentation processes according to the both preceding and current scene descriptions.

These three tasks of CV system are performed differently depending upon time constraints. In the initial training phase, a teacher has to interactively extract texture samples from the first image of a sequence. These samples (attribute vectors) are then provided to the CA&E system along with texture class labels. Because the training phase is usually performed on the first image of a sequence, the initial file of texture concept descriptions can be empty. Acquired initial texture descriptions can be verified on the same image and/or on the next image of a sequence. The resulting descriptions, along with parametric characteristics of system recognition effectiveness, can then be displayed for a teacher. Mutual dialog between a teacher and the integrated system can then be focused on the improvement of

system recognition effectiveness according to teacher-provided evaluation criteria. The objectives of such initial training phase are three fold; i.e.,

(i) to acquire initial descriptions of texture classes from provided image sections of textures (texture samples),
(ii) to improve recognition effectiveness of acquired descriptions according to teacher-provided and domain-dependent evaluation criteria (Pachowicz and Bala, 1991), and
(iii) to verify acquired descriptions through the recognition and segmentation performed on the entire image.

In the second phase (autonomous adaptation), the CV system applies texture descriptions to recognize and segment textured surfaces/objects on the next images of a sequence. The scene understanding module analyzes segmented texture areas along with system recognition effectiveness and compares them with the results obtained on the preceding image(s). At the same time, selected texture samples (image areas representing texture classes) are provided to the CA&E system. The CA&E system monitors the recognition effectiveness of the CV system (i.e., the discriminatory power of texture concept descriptions), and it activates a model evolution task if this effectiveness decreases. The decrease in texture recognition effectiveness carries information about changes in the external perceptual conditions. Such information is necessary to activate concept evolution processes in order to adapt the system to these changes.

This approach assumes that in the training phase texture models are learned from teacher-provided preclassified texture samples. After this, the system has to work autonomously; i.e., without any human supervision. Such autonomous adaptation to a dynamic environment is very difficult and highly dependent on implemented control structures. These control structures and functions must perform the following tasks:

(1) to track system recognition effectiveness over a sequence of images,
(2) to predict future performance of the system,
(3) to indicate those texture classes that have to be evolved; i.e., because their concept descriptions are loosing recognition effectiveness,
(4) to select new training data for the evolution of indicated texture descriptions,
(5) to activate evolution processes, and
(6) to verify effectiveness of evolved models.

The success of the unsupervised adaptation of the CV system to the dynamic changes in the environment necessitates developing new technology that would support all control tasks and the model evolution task. In the following sections of this paper, we provide principles of model evolution, architecture of the CA&E system, evaluation criteria, and experimental results.

2.2. Principles of Model Evolution

The closed-loop cooperation between CV and CA&E systems, illustrated in Figure 1, supports the flow of information necessary for the evolution of texture models over time. This evolution is by *learning from the environment* and it is performed in an incremental mode through the forthcoming images of a sequence. Hence, such an approach to texture recognition requires the implementation of a cooperative and autonomous coexistence of the recognition and learning parts of an integrated system. Such a coexistence can be implemented within a one-level or multi-level control structure. A one-level control structure, presented in Figure 2a, allows adapting the system according to one objective only; i.e., adapting to a sequence of images. On the other hand, a two-level (or multiple-level) hierarchical control structure, presented in Figure 2b, allows adaptation of the system for multiple objectives (for example, both to a sequence of images and to a single image of a sequence).

In a one-level control structure, texture descriptions are applied to recognize texture events on a given image of a sequence. Classified events are then grouped together in order to segment an image into homogeneous areas. Scene understanding processes finalize image segmentation incorporating higher-level reasoning, comparative analysis using preceding images of a sequence, and robot displacement information provided by a robot's hardware. The scene understanding module has a strategic position within the system - it selects texture samples that can be used to evolve texture models. The CV system communicates with the CA&E system in this manner. The modeling and control module of the CA&E system then follows the recognition effectiveness of the integrated system, activating learning processes if they are necessary, and providing new training data for the evolution of object models. Finally, evolved models are stored and searched in order to recognize textures on the next image.

This loop directly integrates the CV system and the CA&E system. Its main goals are: (i) to perceive current system recognition and segmentation capability, (ii) to project future recognition and segmentation effectiveness, and (iii) to provide feedback information and data to the learning module in order to update texture models. For the one-level control structure, the loop also has to select new training data and to activate learning processes. Such a one-level control structure has the capability of updating texture models, however, model verification and multi-objective tune-up cannot directly be secured by the system. These processes can be implemented through this loop if evolved models are

applied again on the same image. Such an approach is computationally very expensive since it involves full vision and learning processes repeated several times on the same image.

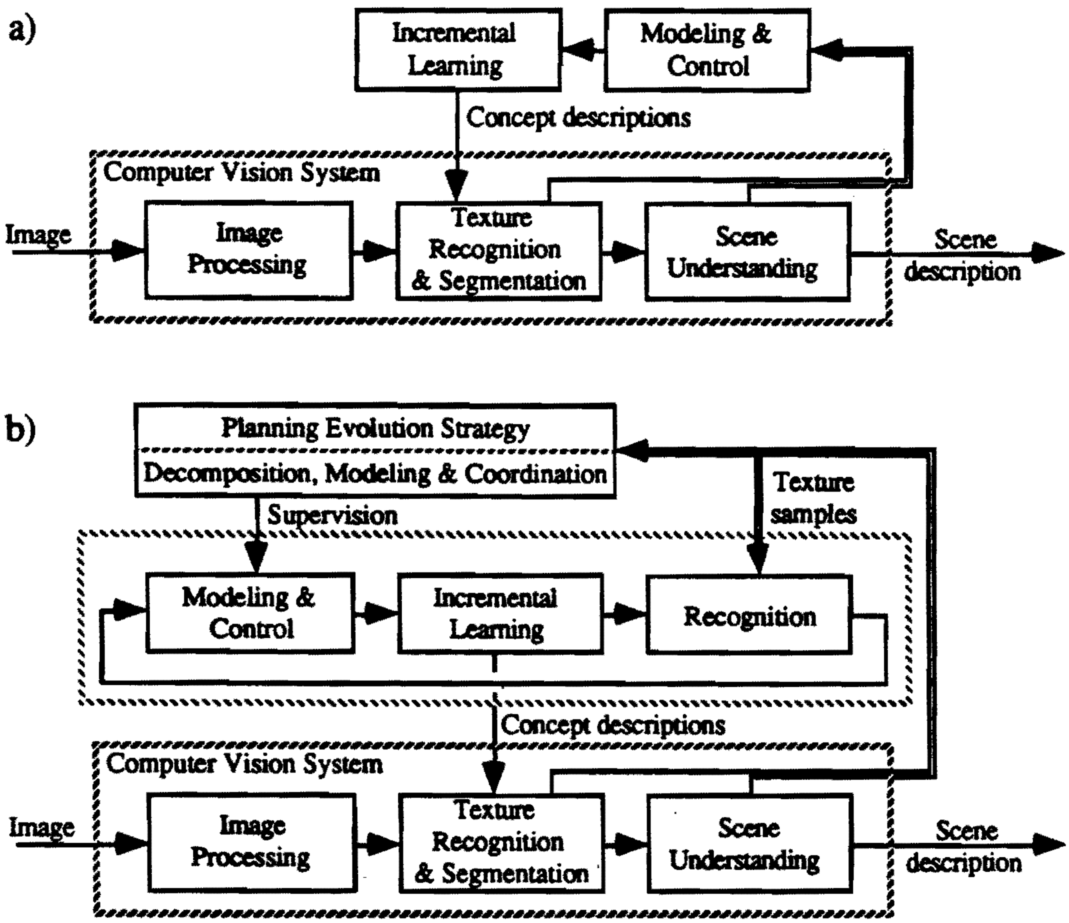

Fig. 2 One-level and two-level control structures for the evolution of texture descriptions

The purpose of model verification and multi-objective tune-up is to secure logical soundness (Rine, 1992), recognition effectiveness, predictive power, and system stability (see section 2.4). System adaptation including these factors can be realized by incorporating a two-level control structure illustrated in Figure 2b. The two loops indicate two control levels of the system. An external loop represents the cooperation of the CV and CA&E systems. This loop supports system adaptation to a sequence of images. The second (internal) loop, supports system adaptation to a single image of a sequence. We expect that the performance of such a multi-level hierarchical system is better than the performance of a system utilizing one-level control structure.

In the two-level control structure, the external control loop follows system recognition and segmentation effectiveness through a sequence of images, it activates evolution processes when needed, and it provides texture samples for the evolution of texture descriptions. The evolution process, however, is managed and executed by the internal control loop of the CA&E system. This loop evolves texture descriptions over a sequence of specially prepared texture subareas called *texture patches*. *Texture patches* of a given class are prepared by cutting texture samples of that class provided by the CV system. Extracted *patches* are then ordered into a sequence (from high to low recognizable *patches*) according to the recognition effectiveness of current texture descriptions on these patches. Local evolution processes are then activated over a sequence of *texture patches*. The goal of such local evolution is (i) to reach higher system stability and (ii) to evolve object models more precisely than by a one-level control structure. Thus, the two-level control structure allows the system to adapt to a single image (over a sequence of *texture patches*) and to a sequence of images.

2.3. Evolution Cycle

Since both control structures support system adaptability in different ways, an evolution cycle is organized differently. For the one-level control structure, an evolution cycle is represented by a single loop integrating the CV system with the CA&E system. This supports system adaptation to a sequence of images. For the two-level control structure, this single loop is decomposed into an additional internal loop of local evolution. This supports system adaptation to a single image of a sequence. These two evolution loops are represented graphically in Figure 3. In the next subsections, we present the functional role of each module taking part in the evolution of concept descriptions along with the integration of these modules into an evolution cycle of the two-level control structure.

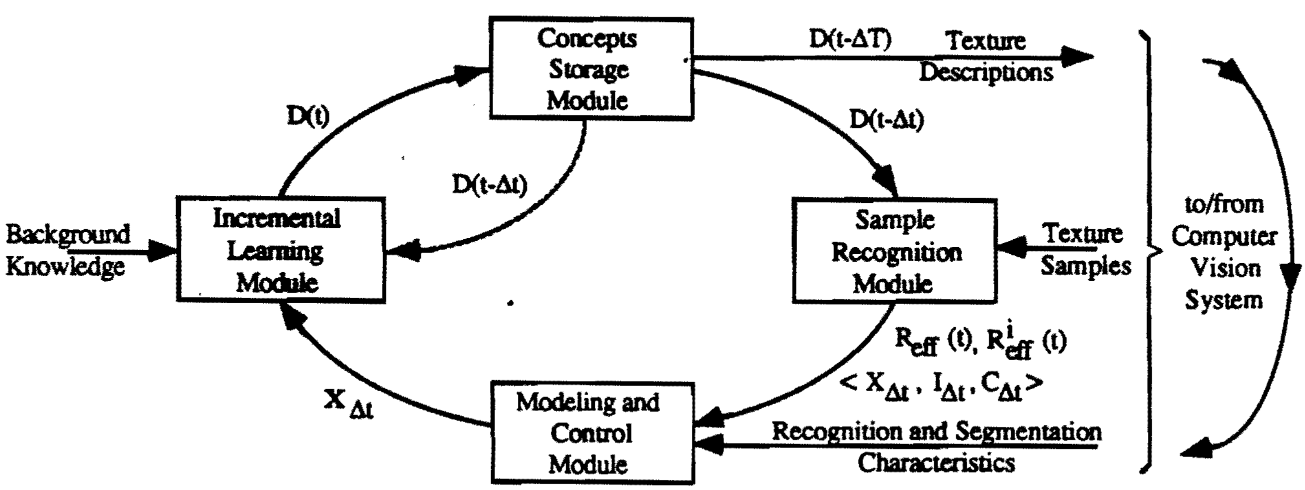

Fig.3 Architecture of the Concept Acquision and Evolution System

2.3.1. Initial Training

As already mentioned above, the purpose of the development of an incremental model evolution approach is the automatic adaptation of a vision system to the dynamic changes in the external environment. Such an automatic adaptation follows an initial training phase that is supervised by a teacher. The goal of this initial training is to teach the system to recognize the most important texture classes typical for a given application domain. The training process is performed on the first image of a sequence; i.e., for the starting position of a robot vehicle. During the training phase, a teacher extracts the most representative texture samples, classifies them into texture classes, and labels these classes by symbolic names. These samples are then provided to the CA&E system and initial concept descriptions are learned.

The L_{t_0} initial learning process of $D^i(t_0)$ initial concept description is repeated for each i-th class of texture drawing inductive inference from the provided $\mathbb{X}_{t_0}$ set of training examples (Michalski, 1983); i.e.,

$$L_{t_0}(\, i, t_0, \mathbb{X}_{t_0,O(i)}, B\,) = D^i(t_0) \tag{1}$$

where

$\mathbb{X}_{t_0,O(i)} = < \mathbb{X}_{t_0,PE(i)}, \mathbb{X}_{t_0,NE(i)} >$ is a pair of $\mathbb{X}_{t_0,PE(i)}$ positive examples and $\mathbb{X}_{t_0,NE(i)}$ negative examples provided to learn initial concept descriptions of i-th class,

$\mathbb{X}_{t_0,PE(i)} = \mathbb{X}^i{}_{t_0}$

$\mathbb{X}_{t_0,NE(i)} = \mathbb{X}_{t_0} - \mathbb{X}^i{}_{t_0}$

$\mathbb{X}_{t_0} = \{\, \mathbb{X}^1{}_{t_0}, \mathbb{X}^2{}_{t_0}, \ldots, \mathbb{X}^{\#classes}{}_{t_0} \,\}$ is a set of initial training examples, and

B represents background knowledge incorporated to perform heuristic search over the space of possible solutions.

The acquired initial concept descriptions are then evaluated and verified on the training image. This approach includes an option for a teacher to optimize acquired concept descriptions to improve recognition effectiveness and system performance according to teacher selected evaluation criteria. These improvements are discussed separately (Pachowicz and Bala, 1991).

2.3.2. Texture Recognition in Dynamic Environments

The recognition process applies learned concept descriptions in order to classify each test texture event; i.e., a given vector of attributes. The goal of the recognition process differs depending upon the level of the system.

The goal of the recognition process performed by the external control loop is (i) to assign class memberships to all texture events of a given image, and (ii) to monitor system recognition effectiveness over a sequence of images. The recognition process R_{CV}, performed by the CV system, incorporates concept descriptions $D(t-\Delta T)$ (previously learned) in order to assign $i \in I$ class membership to each texture event x and to indicate a confidence level $c \in C$ of this decision; i.e.,

$$R_{CV}(t, D(t-\Delta T), x) = \langle i, c \rangle \tag{2}$$

where ΔT represents the time interval of image acquisition between the currently processed image $P(t)$ and the last image $P(t-\Delta T)$ used to evolve concept descriptions. Next, texture segmentation processes unify spatially distributed classification decisions into homogeneous areas of texture classes. Results of image segmentation are analyzed through reasoning processes of scene understanding with respect to the scene contents and the segmentation results obtained for the previous image. Finally, the system recognition effectiveness is calculated for each class of texture represented on the currently processed image.

The results of texture recognition and segmentation phase are transferred from the CV system to the CA&E system. These results include texture samples extracted from the image and characteristics of the system recognition effectiveness (in relation to the past and projected future performance of the system). While recognition effectiveness can be used to plan evolution strategy, texture samples provide information that is necessary to obtain current texture characteristics. This data represents typical areas of texture classes observed from the currently processed image. Texture samples of each class are later cut into smaller sections (*texture patches*) and ordered into a sequence (in decreasing order) according to their recognizability by the $D(t-\Delta T)$ concept descriptions. These *texture patches* are then applied to evolve texture descriptions (by the internal control loop) through the attribute space.

The recognition goal for the internal control loop, the evolution through a sequence of *texture patches*, is (i) to provide guidance for learning processes, and (ii) to verify the effectiveness of the evolution processes and the accurace of concept descriptions. The recognition process $R_{CA\&E}$ is performed in the same way as for the external loop (equation 2) where the most recent concept descriptions $D(t-\Delta t)$ are applied; i.e.,

$$R_{CA\&E}(t, D(t-\Delta t), x) = \langle i, c \rangle \tag{3}$$

where Δt represents the time interval between the current and previous *texture patches* processed within the internal evolution loop. The relationship between the ΔT interval of the external evolution loop and the Δt interval of the internal evolution loop can be expressed by the following equation:

$$\Delta T = \sum_{j=1}^{n} \Delta t_j \tag{4}$$

where n indicates the number of iterations of the internal evolution loop run during a single iteration of the external evolution loop.

The recognition processes of the internal evolution loop are applied on a given consecutive *texture patch* from a sequence of patches prepared from texture samples. The result of such recognition is a set of texture events $X_{\Delta t}$ of a given patch along with associated recognition decisions $I_{\Delta t}$ and confidence levels $C_{\Delta t}$.

Additionally, the overall recognition effectiveness $R_{eff}(t)$ is calculated for the $D(t-\Delta t)$ most recently evolved concept descriptions over texture samples provided by the CV system. This measure along with individual recognition effectiveness $R^i_{eff}(t)$ for each i-th texture class evaluates the effectiveness of applied evolution processes.

2.3.3. Incremental Learning of Texture Descriptions

Since the integrated system has to adapt over time, it has to modify its models dynamically according to new training data. This model modification must be supported by the model acquisition technique working in a dynamic fashion (for example, by the incremental learning methodology).

The incremental learning methodology has already been implemented within several learning programs, i.e., within the AQ family of learning programs (Michalski and Larson, 1978, Michalski and Chilausky, 1989), the INDUCE-4 program (Bentrup, et al., 1987), the ID family of learning programs (Utgoff, 1989), and conceptual clustering (Fisher, 1987, Gennari, et al., 1989). It has been proved that incremental learning increases the speed of learning processes. Unfortunately, this learning technique can give slightly more complex models and somewhat worse recognition effectiveness (Bentrup, et al., 1987, Reinke and Michalski, 1988). The advantages of incremental learning, such as the increase in learning speed and the capability of modifying concept descriptions through new facts, indicate that we can apply this technique to iterative model evolution. Moreover, we apply this learning methodology to the acquisition of texture concepts because it allows (i) modeling a very complex distribution of attributes within the attribute space, (ii) integrating symbolic and numeric attributes within an object model, and (iii) manipulating acquired models in order to optimize them (Pachowicz, 1990, Bala and Pachowicz, 1991, Pachowicz and Bala, 1991).

The new proposed approach to the evolution of texture descriptions incorporates incremental learning in such a way that new training examples $\mathbb{X}_{\Delta t,O(i)}$ must be provided by the system to modify previously learned concept descriptions $D(t\text{-}\Delta t)$; i.e.,

$$L\,(\,i,\,t,\,\mathbb{X}_{\Delta t,O(i)}\,,\,D(t\text{-}\Delta t),\,B\,)\;=\;D^{i}(t) \tag{6}$$

where

$\mathbb{X}_{\Delta t,O(i)} = < \mathbb{X}_{\Delta t,PE(i)}, \mathbb{X}_{\Delta t,NE(i)} >$ is a pair of $\mathbb{X}_{\Delta t,PE(i)}$ positive examples and $\mathbb{X}_{\Delta t,NE(i)}$ negative examples provided as new training data,

$\mathbb{X}_{\Delta t,PE(i)} = \mathbb{X}^{i}_{\Delta t}$,

$\mathbb{X}_{\Delta t,NE(i)} = \mathbb{X}_{\Delta t} - \mathbb{X}^{i}_{\Delta t}$, and

$\mathbb{X}_{\Delta t} = \{\, \mathbb{X}^{1}_{\Delta t}, \mathbb{X}^{2}_{\Delta t}, \ldots, \mathbb{X}^{\#classes}_{\Delta t} \,\}$ is a set of new training data provided for the incremental evolution of concept descriptions.

2.3.4. Integration of Recognition and Learning Processes

While the recognition and learning processes are generally separated, their integration is required in order to adapt the vision system though the evolution of its models. This integration is performed by the Modeling and Control Module (Figure 3) supporting two evolution loops. Therefore, the goal of the integration of recognition and learning processes is split into the following two hierarchically dependent levels; i.e., higher-level integration responsible for system adaptation to a sequence of images, and lower-level integration responsible for system adaptation to a single image.

Since the goal of the higher-level integration is to evolve models over a sequence of images, we distinguish the following tasks of systems integration performed through the external evolution loop:

- to complete and analyze recognition characteristics for each class over a sequence of images (i.e., because recognition characteristics monitor changes of environmental characteristics, they are used to define evolution objectives affected by these changes),
- to plan an *evolution strategy* to be executed by the internal evolution loop (i.e., based on the recognition characteristics and evolution objectives, the system has to generate "a plan" of *evolution actions* - Pachowicz, 1991),
- to control the execution of an *evolution strategy* by the internal evolution loop.

The higher level integration of the recognition and learning modules provides supervision to lower-level integration that directly modifies texture models. This supervision specifies an *evolution strategy*

and decomposes it into a sequence of *evolution actions* to be executed on the lower level. Since the goal of the low-level integration is to adapt texture models to a given image, we distinguish the following tasks that must be performed by the internal evolution loop:

- to select new training data from texture events of a given *patch* to evolve models within a single run of the internal evolution loop,
- to activate learning processes that execute a given *evolution action* as an element of planned *evolution strategy*,
- to verify effectiveness of applied learning processes according to a specified evaluation criteria and performance measures, and
- to monitor and analyze recognition characteristics for each texture class over the internal evolution loops.

The first task, selection of new training data, provides evidences of the dynamic variability of texture characteristics. This data is extracted as a subset $\mathbb{X}_{\Delta t}$ from the set $X_{\Delta t}$ of texture events belonging to a single *texture patch*; i.e.,

$$\mathbb{X}_{\Delta t} \subset X_{\Delta t} \tag{7}$$

This selection process implemented within an experimental system is discussed in sections 3.4 and 5.3, and it is based on the correctness of the recognition decision for each texture event belonging to a given patch. The second task controls the activation of incremental learning module. This activation is performed by means of choosing an *evolution mode* for the learning system (see section 2.5), grouping new training data into positive and negative examples, and defining control parameters for the learning program. The third task verifies evolved models according to previously specified evaluation criteria and performance measures. This verification consists of monitoring the improvement of updated texture models for each interaction of the internal evolution loop. Then, in the fourth task the recognition measurements are collected and analyzed. The fourth task supports the verification of evolution effectiveness of the internal evolution loops; i.e., compares acquired characteristics to expected ones in order to project the effectiveness of the system adaptation to a given image.

2.4. Behavioral factors of model evolution

One can apply input-output behavioral factors typical for control systems to analyze the architecture of the proposed system presented in Figure 2 (Padulo and Arbib, 1974, Jamshidi, 1983, Sastry and Bodson, 1989). These behavioral factors along with AI-oriented behaviors are briefly discussed in

this section in order to define and select guidelines for the investigation and analysis of the model evolution approach (or in general an intelligent autonomous system). Introduced factors include:

- *logical soundness,*
- *controllability, reachability and observability,*
- *stability,*
- *robustness,* and
- *model predictability* (in terms of model discriminating power).

Logical soundness of an autonomous system (Rine, 1992) (an adaptive system) is understood as the system's capability of working within well defined behavioral boundaries (requirements specifications) when the system adapts to the environment. These boundaries define the system allowed behaviors when the system evolves over time, and they protect the system against unpredicted destructive behavior. Any unpredicted behavior of an autonomous system is considered as a negative aspect of system adaptability that has to be in some way accounted for in system requirements specifications. Thus, a logically sound system must always be restricted to a given set of policies (Rine, 1988, 1991, 1992).

An example for restricting the performance of incremental model evolution is the requirement for permanent separation of the initially learned concept descriptions of different texture classes. This requirement protects the system against unplanned merging of two or more concept descriptions of different textures into one concept description. If the concept descriptions of different textures are evolved over the same areas of the attribute space then the system will not be able to distinguish these classes between themselves. In such a case, the scene understanding processes can label and describe the environment incorrectly guiding a system towards unpredicted mixing of class descriptions.

Input-output dynamic behavior of engineering systems is characterized in terms of *controllability, reachability and observability* (Padulo and Arbib, 1974). Considering controllability, we seek an input which will transfer a system from the current state to the zero-state at some future time. Reachability determines whether a system can be transferred from a given state by an appropriate choice of input to the desired state at the desired time. Finally, we say that a given state is observable if it can be distinguished from the other states. From a control theorist's point of view, it is desirable to have a system fulfilling all above characteristics. Real-world systems and particularly AI systems, however, are far away and the identification of system dynamic behaviors is very difficult, complex and sometimes impossible.

Another factor describing the dynamic behaviors of a system is *stability*. Traditionally, if we relate the input to the output, we hope that small changes in initial state eventually have negligible effects upon the behavior of the system. In other words, it is catastrophic for the system if a small perturbation in the input of the system can destabilize (i.e., "blow up") the system input-output behavior. A stable system prevents the negative influence of small perturbations in the input on the output. A system is stable with respect to a given criteria if a system behaves within a given boundary after a perturbation is initiated. Moreover, we say that the best systems behave with respect to the asymptotic borders.

Considering the dynamic input-output behavior of the model evolution system, the highly negative destabilizing effect is observed if learned object models for a given image (of a sequence) are evolved over the following images so that the recognition effectiveness of these models permanently decreases when matched with the same testing data. If models are learned initially then their recognition effectiveness should not deteriorate even if they are incrementally modified in order to adapt them to the dynamic changes of environment characteristics.

This general definition of system stability can be extrapolated towards another objective, i.e., performance consistency. We usually seek the most powerful concept acquisition and recognition methods. These methods can be evaluated by certain measures, for example, the average recognition rate computed through all learned class descriptions. This measure, however, does not define how the system behaves with respect to a particular class. Highly negative effect is when the deviation of recognition rates among individual concepts is large. In such a case, some classes are recognizable with a very high recognition rate but the other classes are recognized very poorly. Logically, we would like to have a system that recognizes each class with a similar confidence level. Then, each class has the same chance to be recognized.

Real-world adaptive systems usually face the following problems (Sastry and Bodson, 1989): (i) there is no detailed state-space model of the system to be controlled, (ii) the system or control is too complex, or (iii) the dynamics of the system is not completely understood. These problems are in scope of system robustness, i.e., the system's ability to maintain stability despite modeling errors and measurement noise. Moreover, robustness deals with situations when an arbitrary small disturbance can destabilize an adaptive system which is otherwise proven to be stable. In that sense, robustness has to consider a disturbance that can occur in extremal future situations that were not investigated during system testing. Complex adaptive systems are not magically robust. They are sensitive to many influences affecting system performance. Practically, we can consider robustness of such systems (i) to output disturbances caused by measurement noise of the output signal, (ii) to unmodelled dynamics caused by system complexity and lack of perfect identification, and (iii) to unpredicted but known classes of input noise influencing system activity. Improving robustness of an

integrated CV and CA&E system presented in section 2.1 is one of major targets for future research. An evolving system working without human supervision has to function properly and this behavior is highly related to the factor of logical soundness discussed above.

Finally, the *predictability* factor determines future system behavior. Since an evolving system working without human supervision extracts new training data for the update of its models, high predictability can (i) improve the accuracy of system evolution, and (ii) increase the speed of system adaptation (because the ΔT time intervals of image acquisition can be larger). The predictability factor affects the system capability of applying (formerly acquired) model descriptions to the next image of a sequence. This factor secures the system against loosing the recognition effectiveness too fast before its model improvement over the next image. The lack of predictability can lead to the incorrect extraction of texture samples used during the model evolution leading to the unsound behavior of a system. Thus, the predictability factor suggests that the discriminating power of object models should be in the wider range of object variable characteristics rather than in the very narrow range.

In this paper, we analyze system stability and some aspects of model predictability within a framework of "partially supervised" evolution of texture models explained in section 3.4.

2.5. Evolution modes

The new proposed approach to the evolution of texture models incorporates incremental learning in such a way that new training examples are provided by the system in order to modify texture descriptions. The modification of texture descriptions is performed by manipulating already existing concept components or creating new concept components in such a way that they cover new positive examples and do not cover any negative example. Such evolution of concept descriptions can be performed under a certain evolution mode. The mode determines the performance of incremental learning processes along with the grouping of new training data and the selection of learning parameters. We distinguish the following three model evolution modes (Pachowicz, 1991): (i) through generalization, (ii) through specialization, and (iii) through generalization-with-specialization.

The first mode, *evolution through generalization*, allows to evolve the model of one class only without influencing other class descriptions. This situation occurs when test data is classified incorrectly and it is not matched strictly with any concept component of a given class. In such a case, test data is provided as positive examples to evolve a given class description through generalization. The class description is then extended by additional concept components covering new positive examples that should have been classified to this class but were not. Thus, the description of a given class is extended through the feature space.

The second mode of model evolution, *evolution through specialization*, evolves the model of a given class by showing negative examples. Negative examples are those test events that were classified to this class incorrectly during the recognition phase and were matched strictly by the description of this class. In such a case, the class description must be modified by shrinking or removing a concept component that covered the test data incorrectly. The learning system can remove the matching concept component if it is small and less typical for the concept. The concept component must be partitioned if it is large or more typical for the concept. One important feature of specialization of class descriptions is that the learning system can manipulate one class with regard to the other classes because it shrinks rather than extends already acquired description.

The third mode of model evolution, *model evolution through generalization-with-specialization*, evolves models of several classes at the same time. While the *model evolution through generalization* and the *evolution through specialization* modify one class description only, the third mode of model evolution applies the same learning data as positive examples to evolve one class description and as negative examples to specialize descriptions of the other classes. When the test data is matched incorrectly to one class and this match is strict, one can evolve this class by applying specialization. But the result of such specialization does not guarantee that the test data will be classified correctly to the second class through flexible matching (i.e., by the shortest distance). Reversing the classification decision, however, is guaranteed when one description is specialized while the second description is generalized at the same time and by the same examples. In such a case, the testing data applied to the learning phase is used to evolve descriptions of both classes.

In this paper, we experiment with adaptive texture recognition incorporating the simplest evolution mode, i.e., *model evolution through generalization*, performed incrementally and under partial supervision (see section 3.4).

3. INVESTIGATING THE INCREMENTAL MODEL GENERALIZATION APPROACH TO THE EVOLUTION OF TEXTURE MODELS

This section presents an application domain of introduced model evolution methodology and discusses its empirical investigation in the area of invariant texture recognition. We define image data, applied attribute extraction method and the evaluation criteria of system performance, and discuss limitations to the primary system structure and autonomy.

3.1. Image Data

The empirical evaluation of the first experiments was applied to the domain of texture using the incremental learning-based evolution of object models. The content of each image was simplified and limited to only a few texture classes. We acquired a sequence of five images presented in Figure 4, where each image consisted of six texture areas corresponding to different sweaters. The images of a sequence were affected both by variable resolution and variable illumination. The resolution was changed gradually beginning from the lowest resolution to the highest resolution; i.e., the progressive movement of a camera. The distance between texture areas and a camera was decreased by half - when distances are compared for the first and the last image of a sequence. The light source was moved along with the camera.

Although the sequence of images was acquired for the decreasing distance from the camera, the nature and dynamics of changes in texture characteristics can differ when one evolves texture models beginning from an image of the highest resolution (towards the image of the lowest resolution), and when such evolution begins from an image of the lowest resolution (towards the image of the highest resolution). Considering these two cases, we decided to carry out experiments for both directions of camera displacement. For each experiment, two sequences of images have been prepared; i.e., a sequence of low-to-high resolution images and a sequence of high-to-low resolution images. The first sequence reflects changes in texture characteristics when the five images are ordered from the lowest to the highest resolution, and the second sequence reflects such changes for the same images but in reverse order.

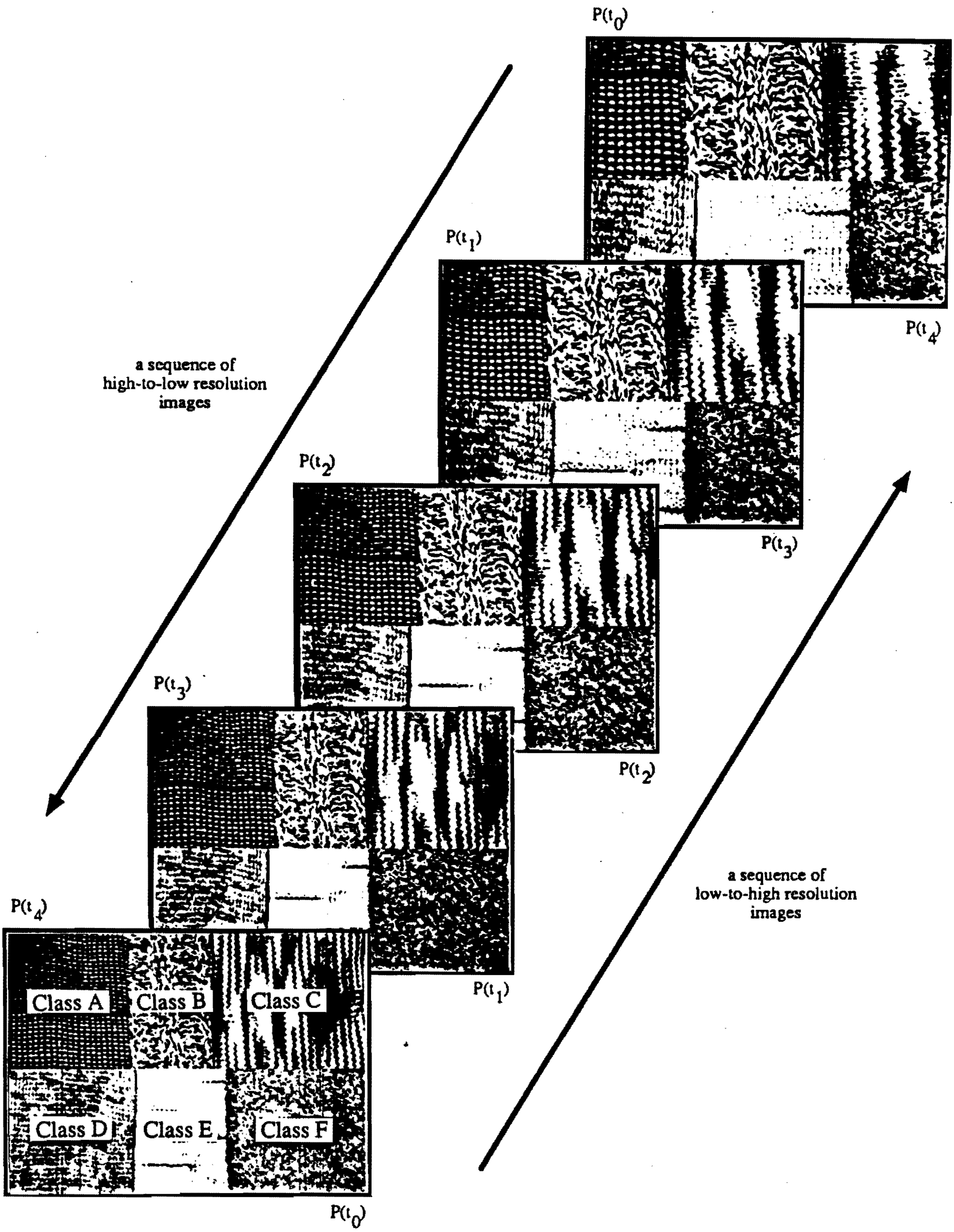

Fig.4 Sequences of images used to investigate incremental learning-based evolution of texture models

3.2. Extraction of Texture Attributes

Each image of a sequence was processed in order to extract texture attributes characterizing texture classes. We modified Laws' (1980) well known method of a two step extraction of texture attributes. In the first step, the local micro-characteristics of raw texture data is computed incorporating specially designed energy masks (filters) detecting local pixel variations over a small window. In the second step, the local macro-statistics are computed in order to derive statistical measures of the filtered image over a larger window. Methods based on similar approach were widely applied (Unser and Eden, 1989, Hsiao and Sawchuk, 1989) to texture feature extraction, and they provided quite good discriminating power when compared with other methods (DuBuf, et al., 1990). Moreover, this class of methods is easily implemented on parallel architectures and these methods are run in real-time.

The modification of the Laws' method of texture feature extraction included: (i) the extension of the number of filtering masks, and (ii) the redesigning of a window used to compute local macro-statistics from the filtered image. The original set of energy masks $M=\{M_i\}$ consists of four 5x5 masks (i.e., R5R5, E5L5, E5S5, and L5S5). This set includes one rotation invariant mask (i.e., R5R5) and three masks that are sensitive to texture directionality (i.e., E5L5, E5S5 and L5S5). We extended the primary list of energy masks by adding three directionality sensitive masks rotated by 90 degrees (i.e., masks: L5E5, S5E5 and S5L5 --- see Figure 5). We also included a 3x3 Laplacian filter (i.e., mask S3S3) to perceive texture roughness of lower resolution. These masks were then applied to transform an input image $f(j,k)$ into a set of images $G=\{g_i\}$ of local micro-characteristics --- through the convolution operation; i.e.,

$$g_i(j,k) = \sum_{m=-a}^{a} \sum_{n=-a}^{a} M_i(m,n)\, f(j+m, k+n) \qquad \text{for } i = 1,\dots,8 \qquad (8)$$

In the second step of feature extraction, Laws (1980) proposed to compute the average absolute value (ABSAVE) over a larger window applied to each filtered image g_i separately; i.e.,

$$h_i(j,k) = 1/(\#S) \sum\sum_{S} | g_i(m,n) | \qquad \text{for } i=1,\dots,8 \qquad (9)$$

where S corresponds to the local averaging window of pixels.

1	-4	6	-4	1
-4	16	-24	16	-4
6	-24	36	-24	6
-4	16	-24	16	-4
1	-4	6	-4	1

R5R5

-1	-4	-6	-4	-1
-2	-8	-12	-8	-2
0	0	0	0	0
2	8	12	8	2
1	4	6	4	1

E5L5

-1	0	2	0	-1
-2	0	4	0	-2
0	0	0	0	0
2	0	-4	0	2
1	0	-2	0	1

E5S5

-1	0	2	0	-1
-4	0	8	0	-4
-6	0	12	0	-6
-4	0	8	0	-4
-1	0	2	0	-1

L5S5

1	-2	1
-2	4	-2
1	-2	1

S3S3

-1	-2	0	2	1
-4	-8	0	8	4
-6	-12	0	12	6
-4	-8	0	8	4
-1	-2	0	2	1

L5E5

-1	-2	0	2	1
0	0	0	0	0
2	4	0	-4	-2
0	0	0	0	0
-1	-2	0	2	1

S5E5

-1	-4	-6	-4	-1
0	0	0	0	0
2	8	12	8	2
0	0	0	0	0
-1	-4	-6	-4	-1

S5L5

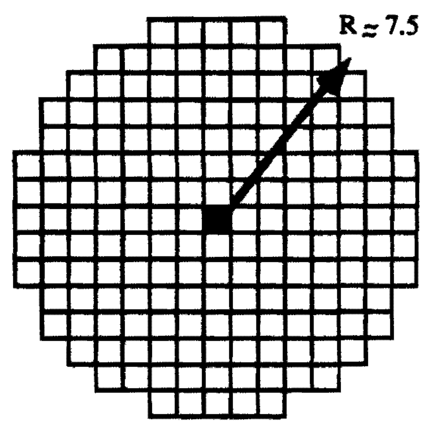

Fig.5 Extended set of Laws' energy masks applied to compute local micro-characteristics, and reshaped averaging window used to compute local macro-statistics of texture energy

In this way, local averaging computes a statistical measure of texture energy. These statistics are considered as a fast approximation to the standard deviation if a zero mean matched filter is applied. The computation of macro-statistics depends both on the shape and size of the moving window, and also on the weight coefficients within the window. We assume that each pixel within the moving window has the same weight equal to 1 , i.e., its influence on the computation of a given feature does not depend on the distance from the center of the moving window.

The large window improves the stability of texture attribute over the homogeneous area. Originally, Laws applied a square moving window of 15x15 pixels. Such a large window, however, causes classification errors when the method is applied to the texture segmentation problem. These errors occur frequently on the borders between different texture areas. Relatively small texture areas are blurred or are even not distinguished at all for a large averaging window. On the other hand, a small averaging window causes extracted texture features to be very noisy which has a significant negative

influence on the acquisition, representation, and recognition of textures. A significant decrease in the size of the moving window has been proposed (Tomita and Tsuji, 1977, Hsiao and Sawchuk, 1989) to deal with the application of the texture energy method to the texture segmentation problem. In this work, we decided to apply a considerably larger averaging window and to reshape it from a square to a circular one of radius $R \approx 7.5$ (see Figure 5). This reshaping stabilizes the distance variation from the central pixel to the border pixels of the averaging window, decreases by 20% the number of pixels taking part in the averaging, and slightly improves the effectiveness of the image segmentation processes.

Acquired texture features are grouped for each (j,k) pixel of an input image into a vector of attributes $h(j,k)=(h_1, h_2, ..., h_8)$, where h_i is a real number greater or equal to zero. A single attribute was then quantized onto 57 levels in order to provide compatible input of training data for the AQ14 symbolic learning program (Pachowicz, 1991). A single texture event was then represented by a vector $x(j,k)=(x_1, x_2, ..., x_8)$, where x_i is an integer in the range from 0 to 56.

3.3. Acquiring Initial Descriptions of Texture

Initial descriptions of texture classes were acquired incorporating texture data extracted from the first image of a sequence. The extraction of initial training data was performed through an interactive process, where a teacher selected image areas corresponding to texture classes. This selection along with class labeling was performed incorporating a specially developed interface between a system and a teacher. Texture samples were then searched randomly within selected image areas to extract a given number of training examples. The number of training examples was assumed to be equal for each texture class; i.e., each texture class was represented by 200 training examples.

The AQ14 learning program was then applied to acquire initial concept descriptions from selected training data. We requested the acquisition of specific rather then general concept descriptions (for the justification see Pachowicz and Bala, 1991). Finally, learned texture class descriptions, represented by rules (i.e., in the Disjunctive Normal Form representation - DNF), were stored within a Concept Storage Module.

3.4. "Partially-Supervised" Evolution of Texture Concept Descriptions

The experiments presented in this paper were carried out in order to demonstrate basic evolution capabilities of the integrated vision and learning system when applied to the texture domain. These first experiments, however, were performed by a simplified system. This system was created based on the general model evolution schema presented in section 2 including the following limitations:

1. texture segmentation was performed by a teacher through the sequence of images,
2. incremental learning of concept descriptions was performed by the AQ14 learning program, and
3. learning processes were initiated every time any texture event were not recognized correctly.

The first limitation, segmentation of texture images by a teacher, was applied in order (i) to secure perfect separation of different texture areas, and (ii) to provide the same number of texture events for their recognition and the selection of new training data. These assumptions allow us to focus on the behavioral analysis of evolution characteristics, where other effects influencing this analysis are eliminated. Perfect segmentation protects the logical soundness of system behavior, excluding the possible merging of two (or more) class descriptions into a single one. Such merging can occur through evolving of one class description with incorrect training data belonging to another class. On the other hand, if the same number of texture events is used to evolve all classes then each class has the same chance to be evolved over the attribute space. Thus, the area of texture class does not dominate the evolution process of one class over another class.

Texture segmentation and the selection of texture events for the evolution experiments were performed through a dialog with a teacher. For each image of a sequence, a teacher pointed out texture areas corresponding to six classes of sweaters. Then, the interface system selected randomly 200 texture events for each class and for each image of a sequence. In this way, we substituted the unsupervised model evolution for a partially-supervised evolution thus securing logical soundness of the system. The experimental work with fully unsupervised evolution of texture models was delayed for future investigation.

Considering the second limitation, we applied the AQ14 learning program to evolve texture models before the creation of a dedicated learning tool. At this point, we have to mention that the AQ14 program, or any other currently existing learning tool, does not fully support model evolution methodology and is not dedicated to handle engineering data while working incrementally. The application of the AQ14 program, however, gave us the necessary experience for the development of a new class of learning tools that will form the kernel of an evolving engineering system.

Finally, the third limitation, the initiation of learning processes by any data that was not recognized, simplified the complexity of control mechanisms that trigger incremental learning processes. The execution of evolution processes is based on the generation of an *evolution strategy* composed of a

sequence of *evolution actions* (section 2.3.4), while triggering the learning tool can include the analysis of both static and dynamic parameters of recognition characteristics. We implemented the simplest *evolution strategy*, and the initiation of learning processes was triggered by the existence of any incorrectly recognized data. When the description of a given class did not correctly recognize a texture event during the recognition phase, the learning process was initiated with this event provided for the incremental learning as a positive example. In this way, the system modified existing concept components or generates an additional concept component(s) to generalize provided positive examples (i.e., applied *evolution action* evolves concept descriptions through generalization only - see section 2.5).

3.5. Evaluation Criteria and Measures of System Performance

The following two major evaluation steps were developed for our experimental system. First, system recognition effectiveness is evaluated when concept descriptions are applied on a single image (static characteristics). Second, time characteristics are acquired and evaluated over a sequence of images.

In this paper, the evaluation of system performance on a single image is based on the following three fold criteria:

1. overall system recognition effectiveness (measured by the average recognition rate),
2. stability of the recognition decision (measured by the standard deviation), and
3. capability of recognizing all texture classes (measured by the minimum recognition rate).

Adapting models to a single image, we would like to obtain the most effective models in terms of recognition effectiveness. Obviously, we are looking for the highest overall performance of these models when applied to a given image used to evolve them. This performance can be expressed by the average recognition rate computed through all six classes of texture. The stability of the recognition decision (measured by the standard deviation) relates to the ideal situation where each class has the same chance to be recognized at the average recognition level. It means that a highly negative effect is observed when one class is recognized with very high confidence (e.g., above 95%) and the other class with relatively low confidence (e.g., below 55%). Finally, we insist that the system must recognize all texture classes it is trained for. Thus, the last criterion can be monitored by the minimum recognition rate through all six classes of texture and should be as high as possible.

Collecting and analyzing time characteristics of system performance, we evaluate the following two factors of the model evolution approach:

1. stability of time characteristics, and
2. predictability of concept descriptions.

Both stability and predictability factors of model evolution secure an evolving engineering system against unsound behavior. Therefore, we expect to have a system of high stability and high predictability.

Stability analysis of time characteristics is oriented towards the investigation of system recognition effectiveness when models are evolved by a given image. These models are then applied to recognize textures of the same image regardless the evolution of these models performed over other images. We would like to have descriptions that are not affected by their evolution through images representing other concept variations. For our application domain, if the system acquired initial concept descriptions from the first image of a sequence, and these descriptions have been evolved through the next four images then the recognition effectiveness of texture descriptions should not be affected by performed evolution processes when applied again to the first image. In that sense, the stability criterion gives the same system recognition effectiveness on the same test data after evolution by another data.

On the other hand, the predictability of concept descriptions applied to recognize textures on the next image of a sequence relates to the recognition effectiveness of acquired or evolved descriptions. Once concept descriptions are evolved by image data, we wish that these concepts are able to recognize textures on the next image. Since the next image of a sequence includes textures perceived under changed perceptual conditions, the effectiveness of applied concept descriptions to the next image can be lower than to the current image. The system gains high model predictability if the decrease in recognition effectiveness is low when models are applied to recognize textures on the next image. This decrease, however, depends also on the ΔT time interval between images of the sequence. This ΔT must be chosen with respect to the dynamics of environment.

3.6. Testing Data and Methodology

A specific testing method was applied to obtain recognition characteristics of evolved texture descriptions in dynamic environments. Considering an objective analysis of system performance, evolution effectiveness must be measured on different sets of data than data used for system evolution. Therefore, testing data used to measure system performance was obtained from different sections of an image than the data used to evolve texture concepts. Since each image of a sequence (Figure 4) is composed of six classes of texture, six testing datasets were obtained from each image. A single dataset of testing data contained 200 testing events characteristic for a single texture class. Considering

a sequence of five images, the total number of testing datasets was equal to 30 datasets (each of 200 testing events), and these datasets were grouped into 5 files corresponding to five images.

The testing data was extracted before running evolution experiments. The testing phase was then applied each time after any evolution loop (i.e., internal or external loop presented in Figure 3). During the testing phase, currently available models were applied to recognize test datasets grouped into five testing files corresponding to five images of a sequence. Thus, recognition characteristics of model evolution were created by collecting the results from each testing phase and each iteration of model evolution. Sections 5 and 6 provide diagrams demonstrating the influence of evolution processes, both for the recognition of textures on each image separately and over consecutive iterations of evolution processes. The presented characteristics represent the recognition effectiveness of evolved models when these models are applied over and over again to the same data during consecutive testing phases.

4. CHARACTERISTICS OF THE LEARNING-BASED TEXTURE RECOGNITION

This section provides introductory characteristics necessary to demonstrate complexity of the application domain, the variability of texture characteristics, and the effectiveness of the learning-based approach to the texture recognition problem. We investigate the distribution of learning data within the attribute space both for a single texture image and for a sequence of images. Next, the choice of a learning approach is discussed and the basic learning characteristics are presented.

4.1. Attribute Space Complexity

The analysis of attribute space complexity has a two fold objective; i.e., (i) to demonstrate the irregularity of attribute distribution, and (ii) to show the variability of texture characteristics over a sequence of images.

The irregularity of attribute distribution is presented through comparison with an approximate normal distribution. Considering multi-dimensional attribute space, the analysis of distribution irregularity is performed for a single attribute. If the distribution is not normal for a single attribute then a multi-dimensional attribute space will not be normal either. Figure 6 presents two examples for such a complex distribution of an attribute, where the solid line corresponds to the smoothed distribution of an attribute and the dotted line corresponds to the approximate normal distribution. Each distribution is presented for a given texture class, for a single image of a sequence, and for a chosen attribute. Both diagrams demonstrate that the distribution of chosen attribute is not normal. Moreover, we found that

many such diagrams show multi-modal distribution (see also Figure 7). Some attribute distributions are considerably flatter demonstrating discriminating weakness of the attribute.

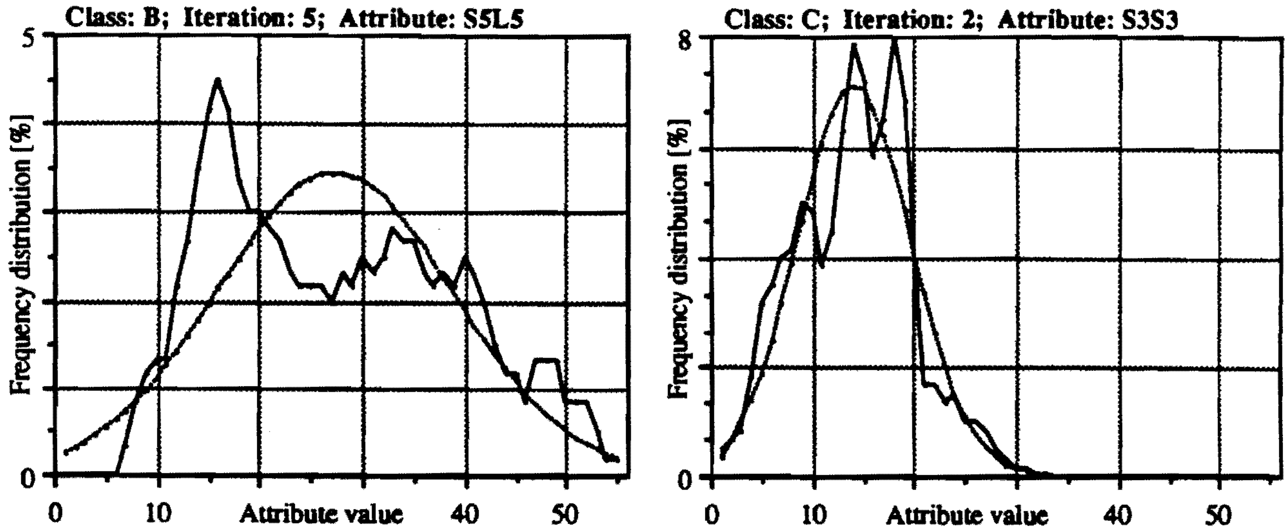

Fig.6 Examples of non-normal attribute distributions

Figure 7 illustrates the variability of attribute distribution through images of a sequence; i.e., under variable perceptual conditions. Two separate cases of attribute distributions are presented in columns of Figure 7. The left column shows the distribution of the L5E5 attribute for the B class of texture and for all five images of a sequence (in order from low to high resolution). It is seen that the attribute has smaller deviation for images of low resolution. And, its significance decreases with increasing the image resolution, showing that an attribute can be more or less distinctive depending on the image resolution. The right column of Figure 7 shows the distribution of the same attribute but for class A of texture. It is seen that the variability of perceptual conditions causes the translation of the attribute distribution within the attribute space.

Both examples for variable distribution of texture attributes demonstrate that an attribute can have a different distribution shape and this distribution can be translated through the attribute space for different image resolutions. These effects cause a degradation of recognition when texture descriptions acquired from the data of one image are applied to another image. This is why we are adapting the vision system to a dynamic environment using the model evolution methodology.

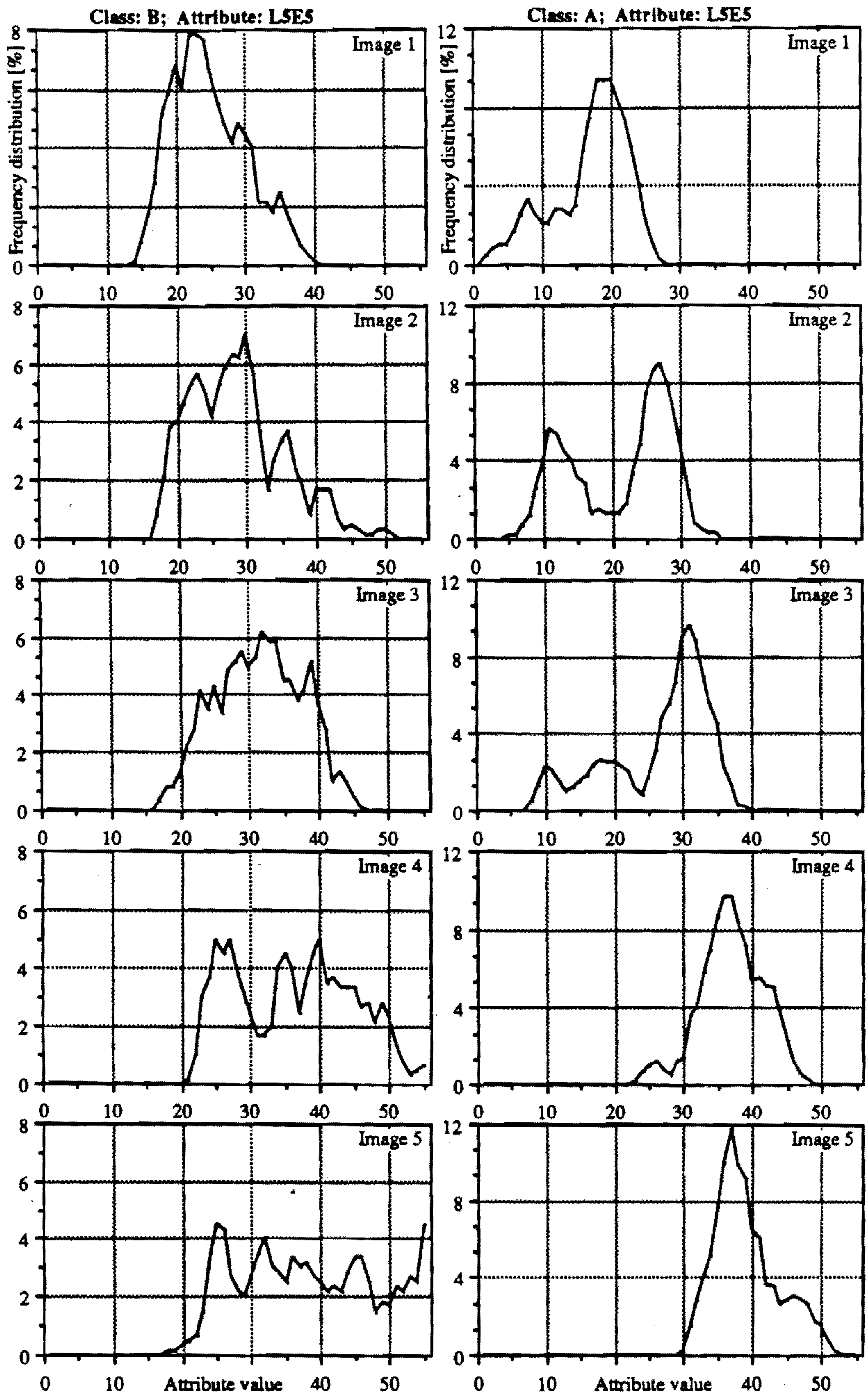

Fig.7 Variability of attribute distribution through images of a sequence --- caused by changing perceptual conditions

4.2. Learning Characteristics

The irregularity and variability of attribute distribution (i.e., texture characteristics) exclude the application of well known parametric pattern recognition methods (Duda and Hart, 1973) to the acquisition and recognition of textures. These powerful statistical methods assume the distribution of attributes to be known (and, particularly, to be normal). Rough approximation of attribute distribution by the normal distribution, however, can sometimes be done successfully if a system learns only a few class descriptions and if object characteristics are significantly different. But, if the system has to learn large number of class descriptions and the attribute space is complex, traditional parametric methods of pattern recognition cannot be applied. The same reason excludes the application of the CAQ learning program (Whitehall, et al., 1990) to the texture recognition problem. The CAQ program deals directly with numeric attributes and is better at removing attribute noise than the AQ14 program, however, it assumes the distribution of training data to be known a-priori and approximated by a parametric distribution.

Investigated basic characteristics of the learning-based texture recognition (incorporated in the AQ14 learning program) include both the dependency of system recognition effectiveness and the complexity of acquired concept descriptions on the number of training examples. In Figure 8, we show both characteristics acquired for the first image of a sequence. Two lines of each diagram correspond to two sequences of texture images; i.e., a sequence of low-to-high resolution images (marked by white circles), and a sequence of high-to-low resolution images (marked by black circles).

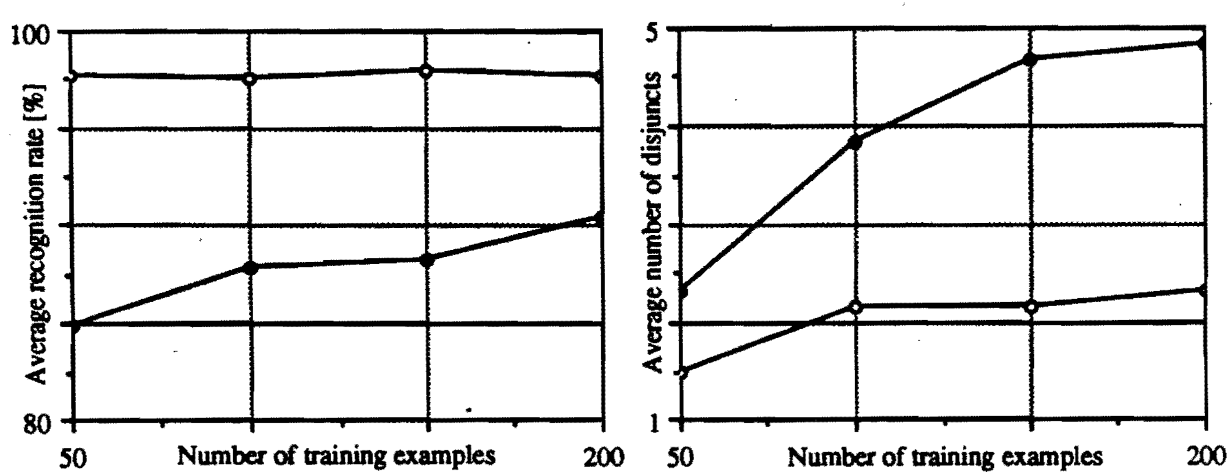

Fig.8 Basic characteristics of learning-based approach to texture recognition (white circles - for first image of low-to-high resolution images, black circles - for first image of high-to-low resolution images)

The characteristics obtained for the first image of the low-to-high resolution images show that the average recognition rate is at a constant and relatively high level through the increasing number of training examples. The increase in the number of training examples has no effect on the improvement of the average recognition rate. In the same manner, average complexity of concept descriptions does not increase significantly. These results suggest that clusters of training data are well separated within the attribute space and the increase in the amount of training data is not necessary.

The characteristics obtained for the first image of the high-to-low resolution images, however, show that an increase in the number of training data improves the average recognition rate. The increase in the average recognition rate is associated with the substantial increase in the average complexity of concept descriptions. Fortunately, this increase is not linear and indicates the piece-wide polynomial characteristic.

After an analysis of learning characteristics, we decided to use 200 examples for each texture sample for our experiments. In this way, we balance the increasing complexity and the recognition effectiveness of learned concept descriptions.

5. EMPIRICAL EVALUATION OF MODEL EVOLUTION APPROACH

The following section presents experimental results from the evolution of texture descriptions incorporating an incremental learning-based evolution approach. The results are provided for two sequences of images, where each sequence was composed of five images of changing resolution and lighting. The recognition characteristics were obtained and analyzed over evolving iterations by monitoring the average recognition rate, standard deviation, minimum recognition rate, and the recognition rate for selected individual classes. Special emphasis was given to the analysis of acquired characteristics with respect to stability and predictability criteria.

Three groups of experiments were run. The first group presents results when the one-level control system was applied to evolve texture models. The second group presents results for the two-level control system. Finally, the third group presents results for the two-level control system integrated with concept optimization and restrictive selection of new training data.

5.1. Evolving Texture Concepts by One-Level Control Structure

The recognition characteristics (i.e., average recognition rate, standard deviation, and the minimum recognition rate) obtained for the one-level control system are presented in Figure 9 for both sequences

of images (see columns of Figure 9). Each diagram contains recognition characteristics for testing datasets of five images and through five iteration steps.

A single curve monitors the recognition performance of evolved texture descriptions on a given image (of a sequence) over consecutive evolution iterations. Let us consider the recognition effectiveness, for example, of models initially acquired from the first image of a sequence. Once acquired models are then evolved over the next images of the sequence; i.e., by the second, third, fourth and fifth image. Every time the models are updated, the system measures their recognition performance on the same testing image; i.e., in our example on the first image. Completed characteristics show the recognition performance of evolved models respective to the first image only; i.e., we can analyze recognition stability of the system.

On the other hand, models evolved by preceding images of a sequence can be tested on a particular image that follows these images. For example, models evolved by the first, second, third, and fourth image of a sequence can be applied to the fifth image after each evolution iteration. It means that we measure system recognition performance on a future image of the sequence; i.e., we can investigate the predictive power of texture models. Finally, models are evolved by the fifth image and the effect of their adaptation can be tested as well.

Characteristics obtained for both image sequences show unstable performance in regard to recognition stability of the evolving system. A lack of stability is seen particularly for the first sequence of images, i.e., for images from low-to-high resolution. Two negative evolution effects are observed that cause recognition instability. The first negative effect, so called *lack-of-support*, is explained graphically on a hypothetical curve in Figure 10a, and it is seen for the 1st image characteristic in Figure 9. The *lack-of-support* relates to the decrease in the recognition effectiveness for a given image when the evolution of models is continued through the next images --- see the reference position and analysis direction in Figure 10. We expect that object models once acquired will have similar recognition effectiveness despite they are evolved over other images characterizing object variable occurrences.

The second negative effect, so called *lack-of-progress*, is graphically illustrated on a hypothetical curve in Figure 10b, and it is seen for the 5th image characteristic in Figure 9. The *lack-of-progress* relates to the unexpected drop of the recognition effectiveness after the adaptation is performed to a given image. In this case, we expect that the recognition effectiveness will be higher (or at least on the similar level) for the current referral image rather than for the directly preceding one.

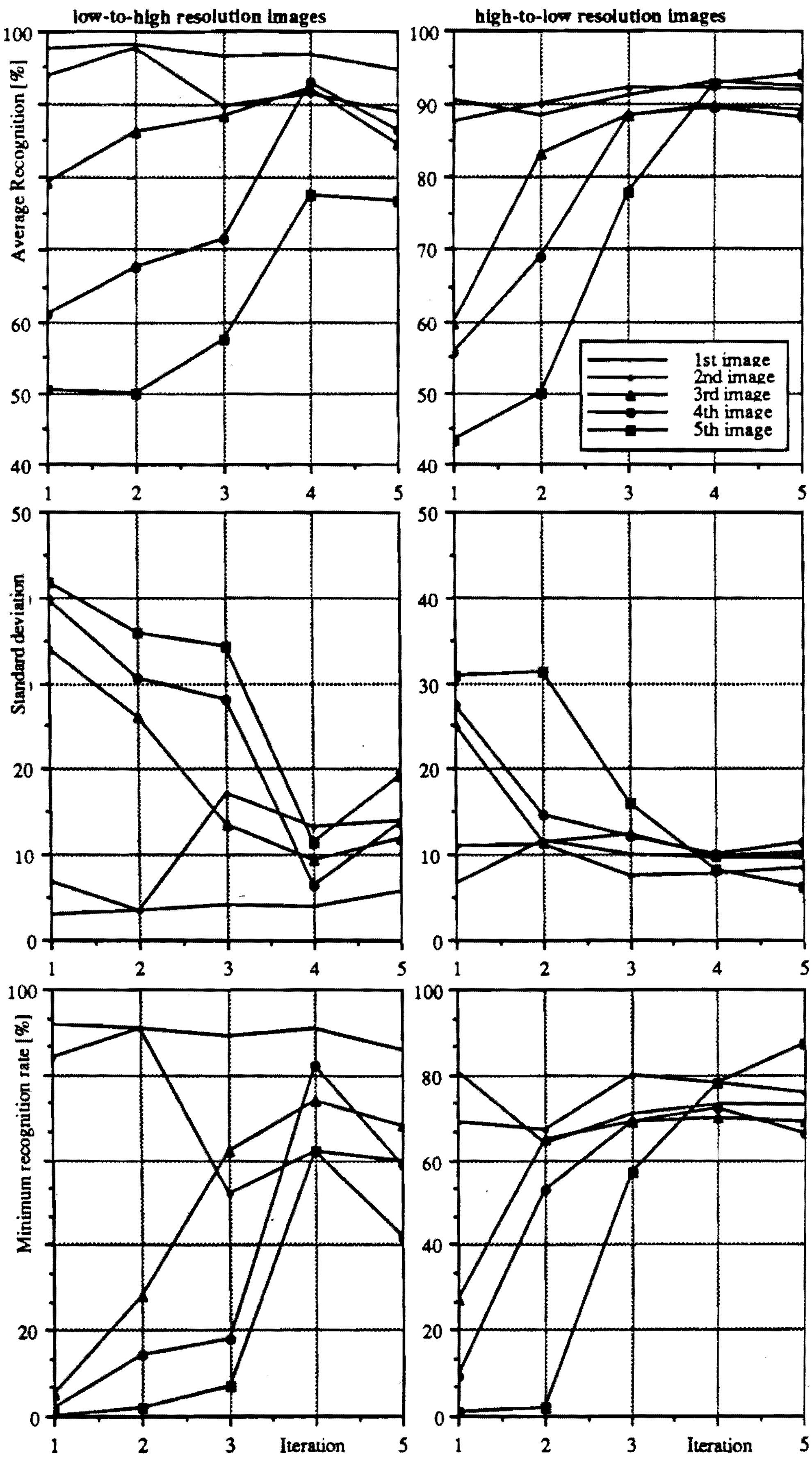

Fig.9 Characteristics of system recognition effectiveness for the evolution of texture concepts by one-level control structure

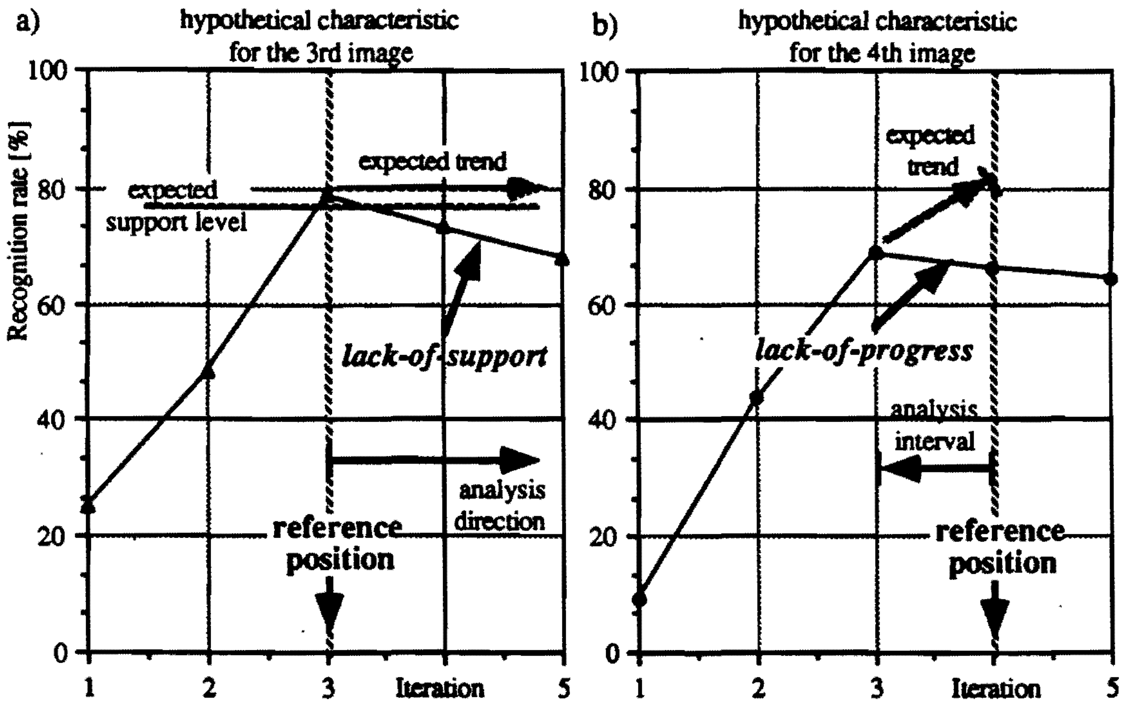

Fig.10 Visual explanation of two negative effects causing recognition instability (i.e., (a) *lack-of-support* and (b) *lack-of-progress*) - the reference position relates to the position (n-th) of an image within a sequence, the analysis direction depends on which effect is analyzed

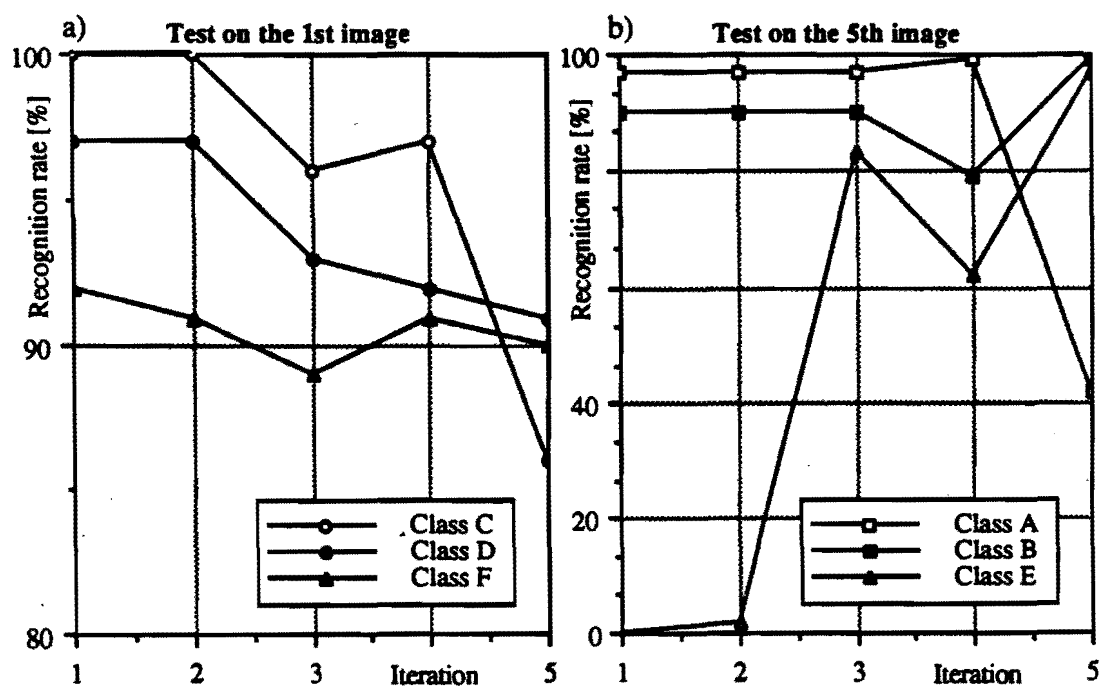

Fig.11 Instability of recognition characteristics for the low-to-high resolution images

Both these negative effects are better illustrated in Figure 11a and Figure 11b, respectively. These figures present selected individual recognition characteristics of three classes which influence mentioned instability of system recognition.

Figure 11a illustrates the deteriorating recognition effectiveness of initially acquired descriptions of three texture classes (i.e., classes C, D and F) when these descriptions are evolved over the next images. As mentioned before, we would like to have models such that they adapt once to a given image and then do not loose their discriminating power during the evolution over subsequent images. If models are sensitive to the evolution iterations that adapt the system to concept variations then a large number of such iterations can weaken their discriminating power. This could make the system unable to recognize the concept variation learned previously. We find that *lack-of-support* for recognition effectiveness is caused by noise accumulation over the sequence of images. Noise accumulation occurs because texture data is noisy. In the current system, the selection of new training data does not exclude noisy examples from being provided to the learning module. This selection of new training data was performed by the simplest method and limited to the extraction of all texture events that were not recognized. In this way, new training datasets of selected texture events are characterized by lower Signal-to-Noise Ratio than datasets provided to the recognition phase. More careful selection of new training data is recommended and it was applied in experiments presented in section 5.3.

Details of the second negative effect of model evolution, i.e. *lack-of-progress*, are presented in Figure 11b. In this case, evolved models were applied over and over again to recognize textures on the last image of the sequence; i.e., on the 5th image. It is seen that the predictive power of the description of class A was very high from the first to the fourth iteration. This high prediction caused that the description of class A was not evolved over preceding iterations (i.e., over the first, second, third and fourth images). In this way, the model of class A did not compete with other models. Finally, it was not evolved over the fifth iteration (i.e., over the fifth image of a sequence) due to the empty set of new training data for class A --- class A was recognized perfectly by models after the fourth evolution iteration. In the same time, descriptions of classes B and E were aggressively evolved during the last iteration, eroding the description of class A. Classes B and E were recognized at a much lower recognition level in the preceding iterations, causing the new training datasets for these classes to consist of data that significantly moved concept descriptions through the attribute space. Two solutions to the elimination of this negative effect (causing recognition instability) were proposed; i.e., (i) the development of a two-level control system that evolves models through split training datasets (see next section), and (ii) the design of a new evolution-oriented learning kernel that allows for continuous evolution of model descriptions.

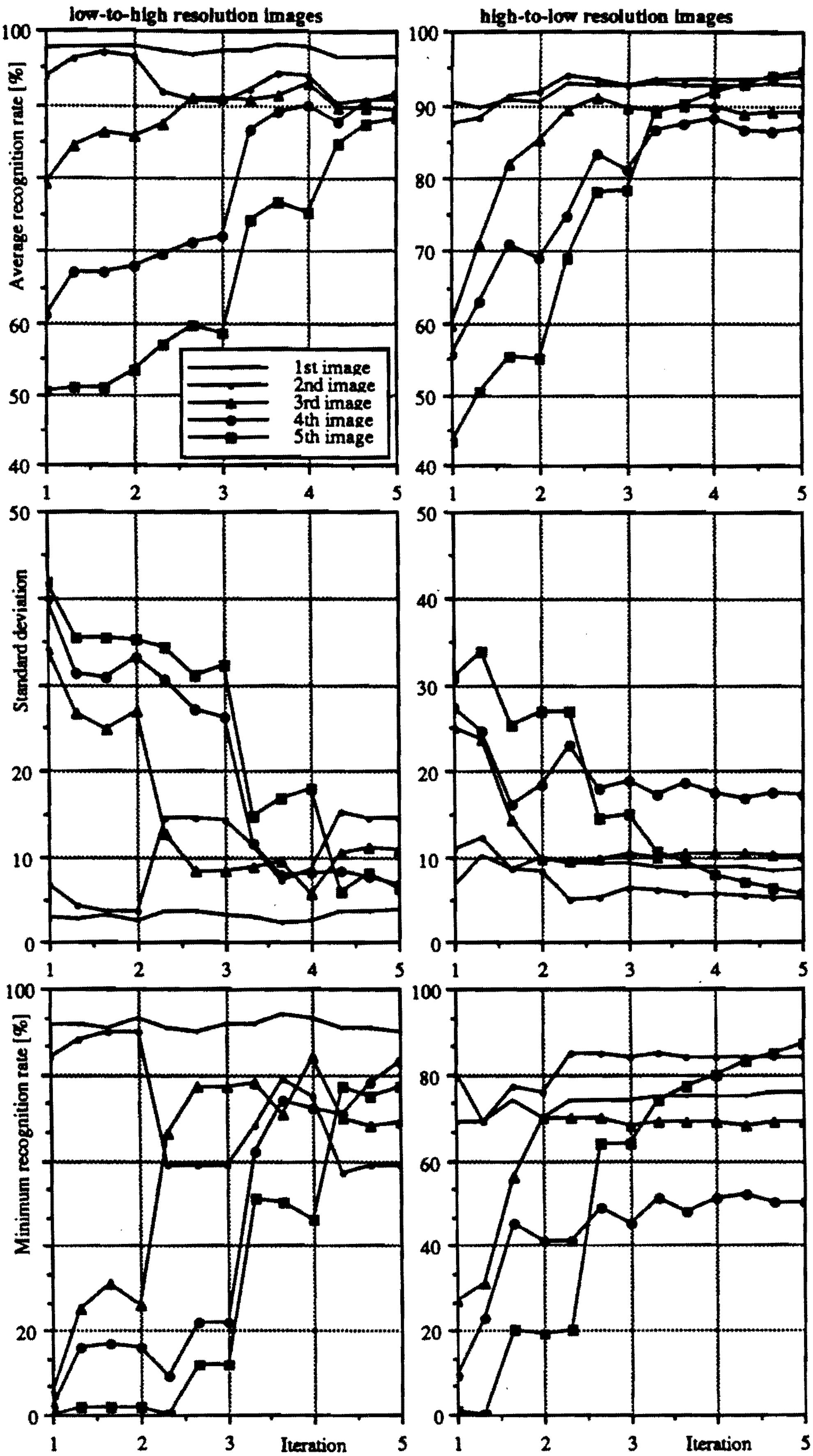

Fig.12 Characteristics of system recognition effectiveness for the evolution of texture concepts by two-level control structure

5.2. Evolving Texture Concepts by the Two-Level Control Structure

Recognition characteristics are presented in Figure 12 of applied model evolution approach that incorporates two-level control structure (where the two columns correspond to two sequences of images). These characteristics show two types of measurement points; i.e., points corresponding to the external evolution loop (see integer iteration values), and points corresponding to the activated internal evolution loop (see fractional iteration values). There were five iterations of the external evolution loop. For each external loop there were three iterations of the internal evolution loop.

The evolution of texture models incorporating two-level control structure reduces the two negative effects that cause recognition instability, i.e. *lack-of-support* and *lack-of-progress*. Details of the improvement of system recognition stability are illustrated in Figure 13a and Figure 13b, respectively. The left diagram presents slight improvement in the support for discriminating power of chosen concept descriptions. The *lack-of-support*, however, has not been eliminated completely. On the other hand, the right diagram presents significant reduction of the *lack-of-progress* effect (when compared with the diagram presented in Figure 11b). The model evolution through the internal loop even recovered the recognition effectiveness for the description of class A. This recognition effectiveness did not drop to a dangerous low level indicating that class A was not recognized at all.

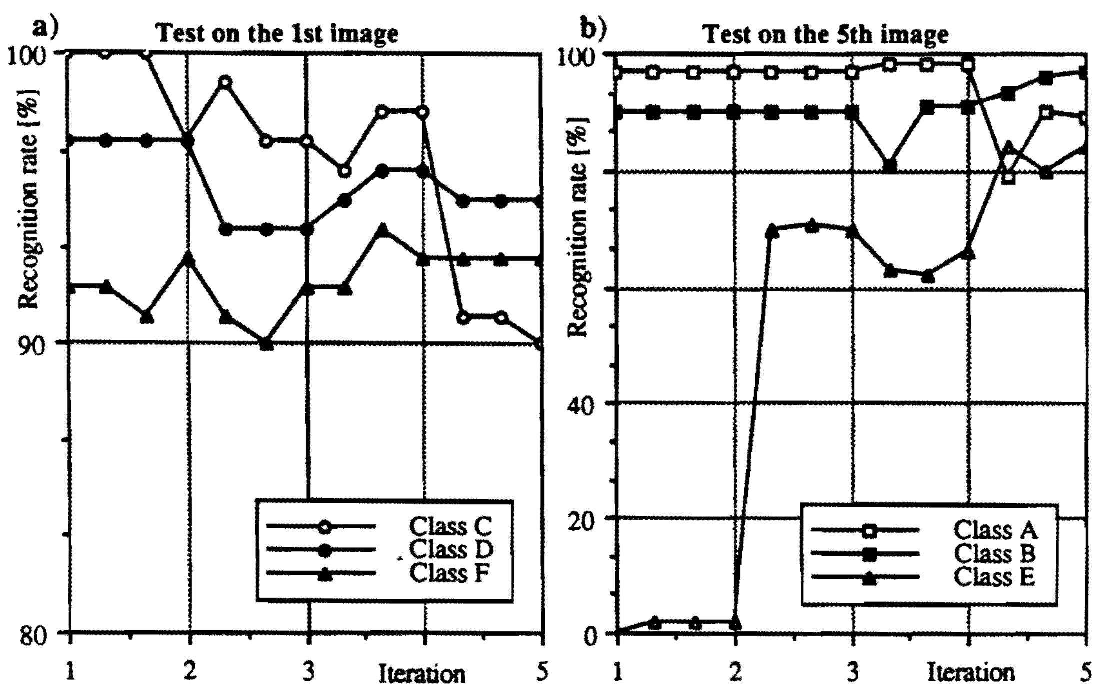

Fig.13 Improvement of system recognition stability reached by the two-level control structure

The evolution of texture models by the two-level system also improves the general stability of the system that is demonstrated by smoother recognition curves. A new negative side-effect, however, is found when analyzing diagrams presenting the minimum recognition rate. For the high-to-low resolution images, the minimum recognition rate through six texture classes is decreased significantly to the 50% level. We will follow this drop in the minimum recognition rate in the third set of experiments, presented in the next section.

5.3. Integrating Model Evolution with Data Filtering

The last group of evolution experiments was done in order to demonstrate the effectiveness of the model evolution approach integrated with noise reduction. The two-level control structure of the system was improved by (i) the optimization of initially acquired texture concepts, and (ii) the filtration of new training data provided for each evolution iteration.

Initially acquired concept descriptions from the first image of a sequence were optimized in order to reduce the influence of noise on the initial concept descriptions. We applied an already developed concept optimization method (Pachowicz and Bala, 1991) dedicated to learn from noisy and complex engineering data. Concept descriptions were then evolved incorporating new training data in the same way as indicated in the previous section. The critical point of such evolution, however, is that the new training data used to evolve texture descriptions is also noisy. If we consider the Signal-to-Noise Ratio (SNR) for a new training dataset as a subset of the primary texture sample, this rate is lower than for the primary texture sample used to select this new training dataset. This situation creates an enormous problem with noise accumulation over evolution cycles.

The reduction of noise in the evolved concept descriptions can be two-fold; i.e., (i) through the elimination of noisy examples from a new training dataset, and (ii) through the elimination of noisy concept components from evolved descriptions. Since the elimination of noisy concept components from evolved descriptions is a very difficult task, we applied very simple filtering of new training datasets in order to decease the influence of noise on the evolution processes. This filtering removed a given percentage number (i.e., 10% in our experiments) of the "worst" examples from a new training dataset. Applied criterion for the evaluation of each example was the number of nearest neighbors (of the same class) in the attribute space within a given distance from the evaluated event.

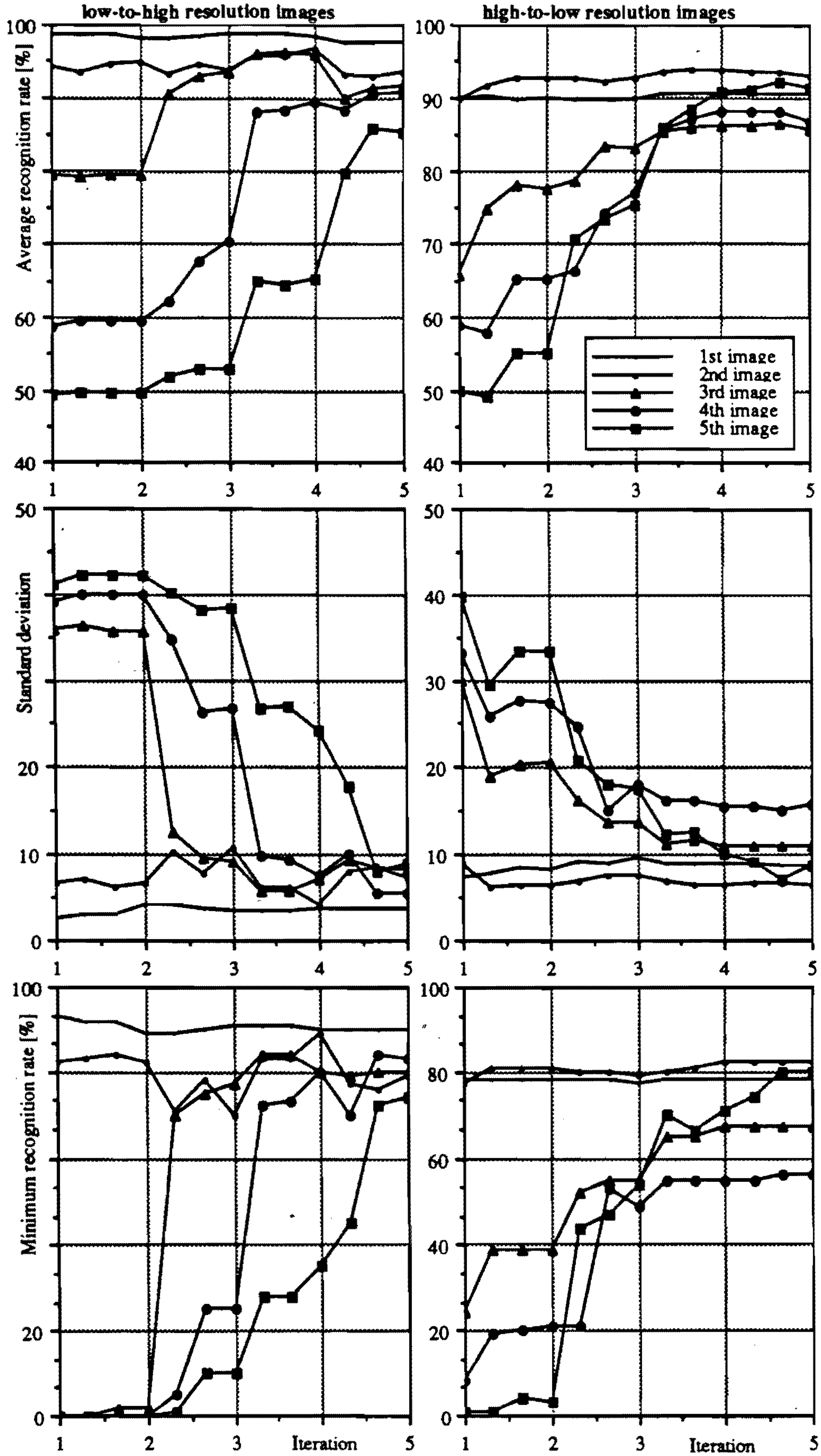

Fig.14 Characteristics of system recognition effectiveness for the evolution of texture concepts by integrated two-level control structure and data filtering

The characteristics are presented in Figure 14 obtained for this model evolution approach integrating two-level control structure with data filtering. These characteristics demonstrate further improvement in recognition stability substantially reducing both negative effects; i.e., *lack-of-support* and *lack-of-progress*. The negative side-effect mentioned in section 5.2 occurring when analyzing the minimum recognition rate has also been slightly reduced.

Details of the improvement in the recognition stability can be seen in Figure 15, where individual recognition characteristics for selected classes are presented. This figure can be compared with Figures 11 and 13 presenting characteristics of the same classes and for the same test image but for different versions of system architecture. Both negative effects causing recognition instability were successfully eliminated through the improvement of the evolution schema and system architecture. The cost of these modifications is seen in Figure 15b and it relates to an increase in the deterioration of predictive power of concept descriptions --- for a discussion of this side effect see the next section.

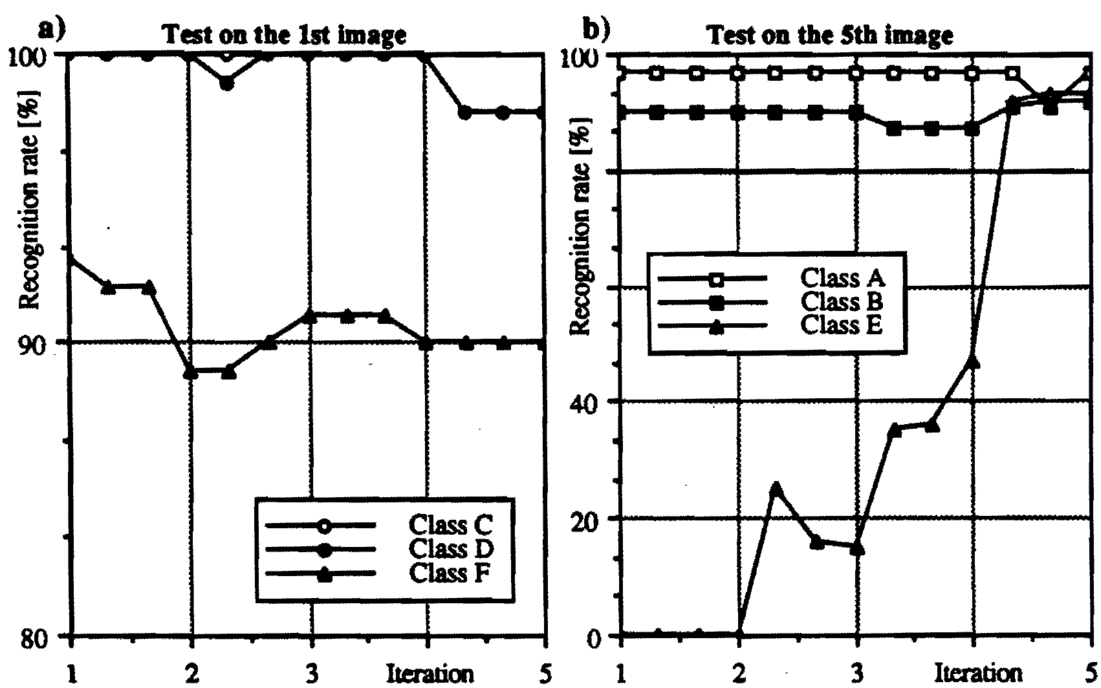

Fig.15 Improvement of system recognition stability reached by the two-level control structure integrated with data filtering

6. SUMMARY OF RESULTS

As a summary of the experiments explained above, we present a comparison of results as a judgement of applied methodology, developed system architectures and modifications. This comparative study is

provided in terms of concept complexity, recognition stability, and predictive power of evolved models.

Characteristics of the average complexity of evolved texture concepts are presented in Figure 16 for both sequences of images; i.e., for low-to-high resolution images, and for high-to-low resolution images. Both diagrams show almost linear growth of concept complexity along with the number of evolution iterations. The lowest complexity, however, has been regularly monitored for the third series of experiments; i.e., where the model evolution incorporates two-level control architecture integrated with the filtration of new training data. The complexity of texture concepts was the highest for the two-level control architecture indicating the highest possible rate of noise accumulation during evolution.

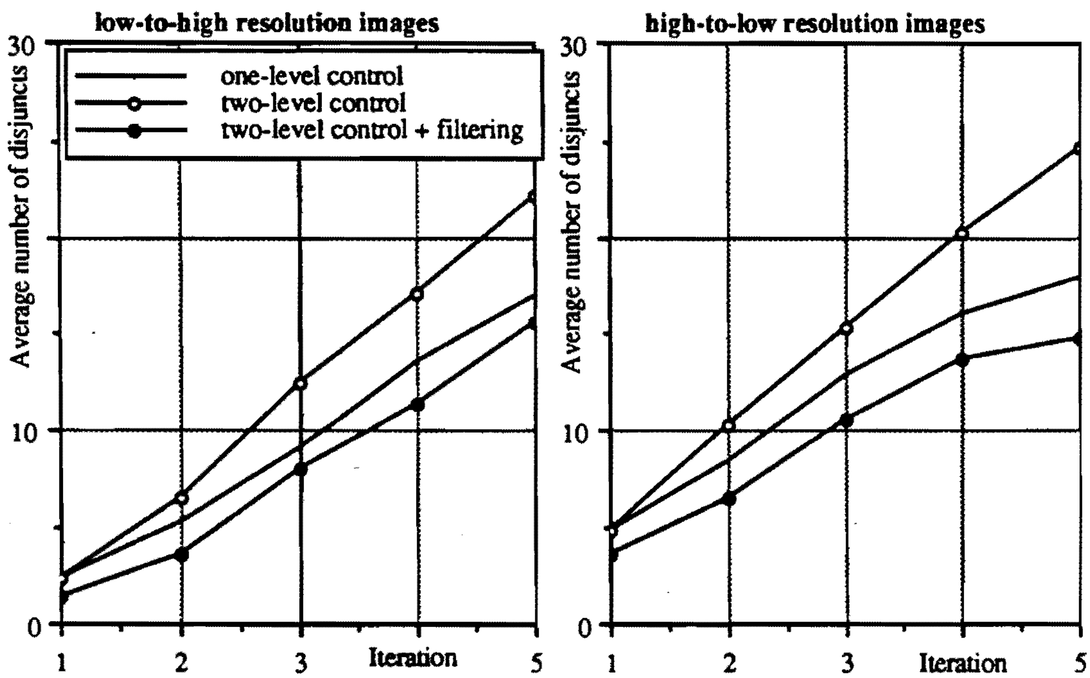

Fig.16 Complexity of evolved texture descriptions

Figures 17 and 18 compare the *lack-of-support* and *lack-of-progress* as the recognition instability effect for three different versions of sy$tem architecture. The best results are acquired for the two-level control structure with data filtering. Both negative effects have been reduced significantly. Moreover, the recognition characteristics are smoothed when compared with characteristics acquired for the other two system structures.

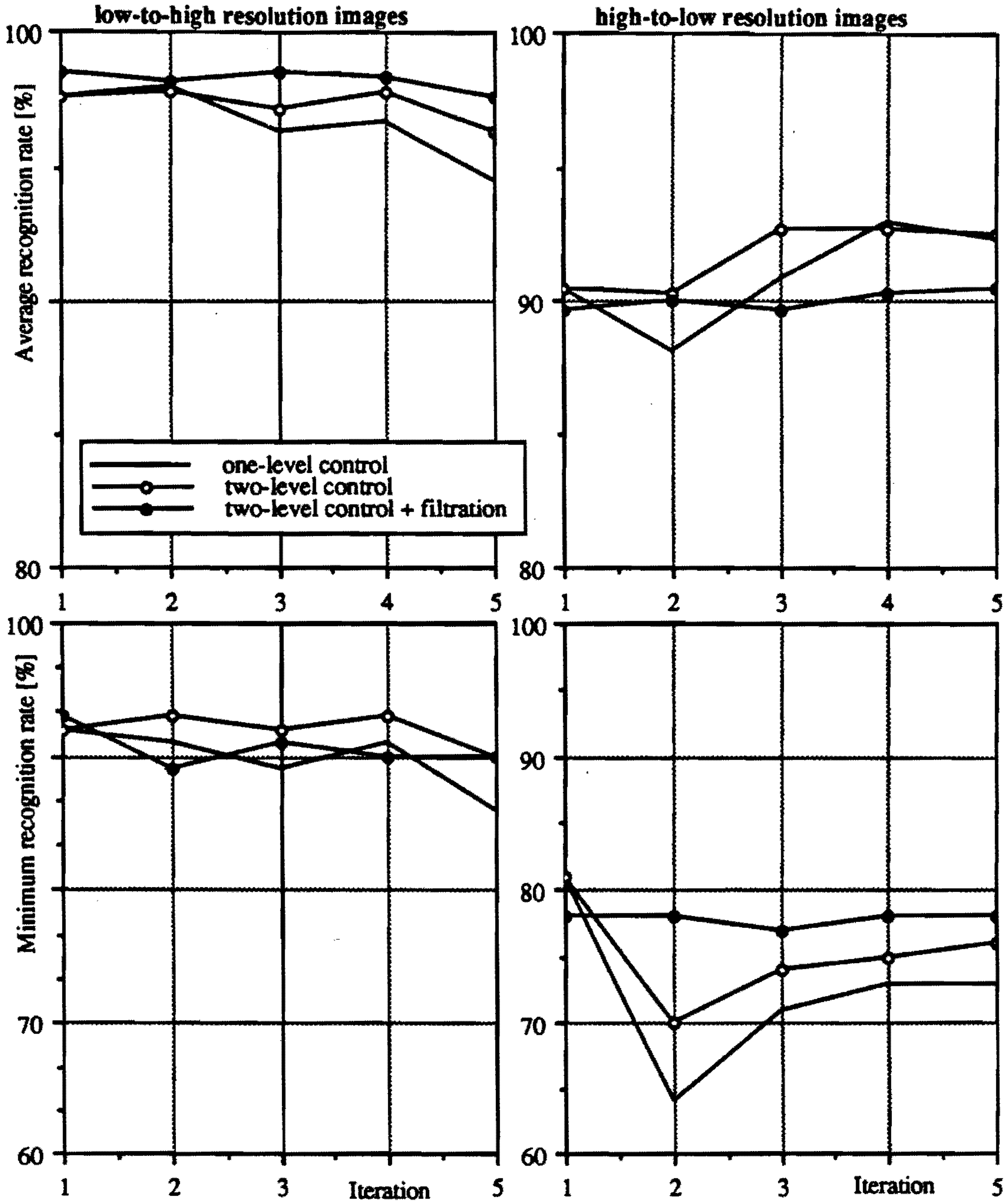

Fig.17 Summary of results for the first image of a sequence

Finally, we compare the predictive power of the concept descriptions as applied to the analysis of recognition characteristics for the last image (fifth image) of a sequence --- see Figure 18. Unfortunately, we find that this prediction power is lower for the third group of experiments; i.e., for the two-level control with data filtering. After fourth iterations (i.e., after the first, second, third and fourth image), textures of the fifth image are recognized on much lower recognition level, where the the average recognition rate drops from about 75% to about 65%. It is seen that the characteristics

are shifted. This drop, however, does not have a significant influence on the fifth evolution iteration and the average recognition rate is recovered substantially.

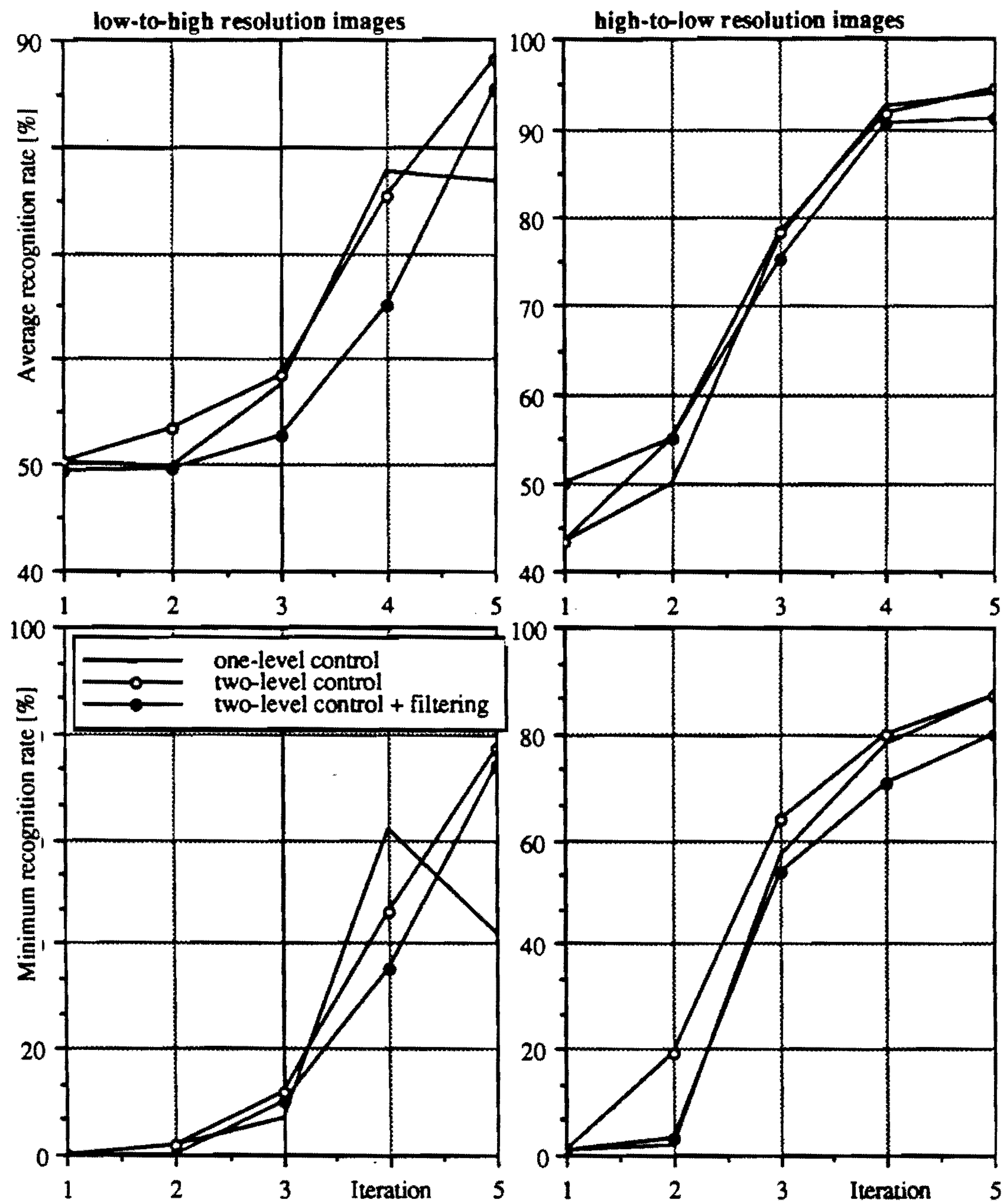

Fig.18 Summary of results for the last image of a sequence

The lower prediction power of concept descriptions is caused by the following two reasons; i.e., (i) the filtration of new training data decreases the number of training examples, and (ii) texture

descriptions are not evolved continuously through the attribute space. First of all, the dependency of the recognition effectiveness on the number of training data has been presented in Figure 8. It shows that the recognition effectiveness of concept descriptions depends upon the number of training data. Since the filtration removes some noisy and some correct training examples the recognition effectiveness can be decreased. Moreover, images of a sequence are acquired for certain time intervals and monitor step changes rather than continuous changes of perceptual conditions. Applied incremental learning (AQ14 program) then has to update concept descriptions creating new concept components rather than modifying the old ones. This is because the difference in perceptual conditions is very significant --- see the variability of attribute distribution through images of a sequence presented in Figure 7 and the increasing complexity of concept descriptions presented in Figure 16. The deteriorating predictability of concept descriptions observed for the integrated two-level control structure with data filtering can be eliminated in the future by the creation of a dedicated model evolution tool that will replace the AQ14 learning program.

7. CONCLUSIONS

The paper has presented a new approach to the recognition of objects in dynamic environments (where objects have variable characteristics) through the evolution of their models over time and without human supervision. The approach integrates computer vision with machine learning and assumes that

- the system has to recognize objects on each image of a sequence,
- the images demonstrate the variability of conditions under which objects are perceived,
- an observer and objects can move,
- the extraction of texture attributes and training examples can be imperfect, and
- the system has to work autonomously (i.e., without teacher help).

This integration of computer vision and machine learning was implemented in order to perform system evolution in a dynamic fashion which is crucial for the development of autonomous intelligent systems. We incorporated images of a sequence to adapt system models to perceived variabilities of texture characteristics. Such an adaptation integrates the recognition and segmentation processes of computer vision with the incremental knowledge acquisition processes of machine learning. While the initial acquisition of texture models is driven by a teacher, the evolution of these models is performed over a sequence of images without teacher help. The texture descriptions initially acquired are applied to recognize and to extract objects on the next images. The effectiveness of such recognition and object extraction is monitored. When this effectiveness decreases, the system selects new training data and it activates learning processes in order to improve its models.

The experiments presented in this paper were run while limiting fully autonomous model evolution to "partially-supervised" evolution. The experiments were compared based on the following three system configurations: (i) a one-level control structure, (ii) a two-level control structure, and (iii) a two-level control structure with data filtering. Obtained results were evaluated investigating system recognition effectiveness, system stability, and the predictive power of the evolved models.

In the scope of current and future work is the continuation of interdisciplinary research leading towards the creation of the next generation of an integrated vision and learning system that is an intelligent autonomous system. This research is continued to (i) improve computer vision, with special emphasis on the recognition, segmentation and high-level reasoning about segmented images, (ii) develop a new class of learning tools supporting incremental model evolution in different modes, (iii) develop control strategies for models evolution, and (iv) elaborate a theoretical basis for the investigation of evolving systems and measurement of their performance.

Future experiments include further modifications of presented system architectures. These modifications include:

- elimination of partial supervision through the development of a scene understanding module for automatic selection of texture samples,
- more precise selection of new training data, and
- precise guidance of the evolution of concept descriptions.

The next experiments will also include the investigation of model evolution methodology incorporating two other evolution modes; i.e., evolution through concept specialization, and evolution through generalization-and-specialization.

ACKNOWLEDGEMENT

Author wish to thank David Rine and Eugene Norris for constructive comments, Michael Hieb and Pervez Mohta for help in the preparation of this paper. This research was done in the GMU Center for Artificial Intelligence. Research of the Center is supported in part by the Defense Advanced Research Projects Agency under the grant administered by the Office of Naval Research No. N00014-87-K-0874 and No. N00014-91-J-1854, in part by the Office of Naval Research under grants No. N00014-88-K-0397, No. N00014-88-K-0226 and No. N00014-91-J-1351, and in part by the National Science Foundation Grant No. IRI-9020266.

8. REFERENCES

Arkin, R. C., "Integrating Behavioral, Perceptual, and World Knowledge in Reactive Navigation", *Robotics and Autonomous Systems*, Vol.6, pp.105-122, 1990.

Bala, J.W. and P.W. Pachowicz, "Recognition of Noisy and Imperfect Texture Concepts via Iterative Optimization of Their Rule Descriptions", submitted to *Int. J. of Pattern Recognition and Artificial Intelligence*, 1990.

Bentrup, J.A., G.J. Mehler and J.D. Riedesel, "INDUCE 4: A Program for Incrementally Learning Structural Descriptions from Examples", UIUCDCS-F-87-958, Computer Science Department, University of Illinois, Urbana, 1987.

Bhanu, B., S. Lee and J. Ming, "Adaptive Image Segmentation Using a Genetic Algorithm", Proc. Image Understanding Workshop, Palo Alto, CA, pp.1043-1055, 1989.

Bhanu, B., S. Lee and J. Ming, "Self-Optimizing Control System for Adaptive Image Segmentation", Pittsburg, PA, pp.583-596, 1990.

Brooks, R., "The Whole Iguana", in Robotics Science, M. Brady (Ed.), The MIT Press, pp. 432-458, 1989.

Cross, G.R. and A.K. Jain, "Markov Random Field Texture Models", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol.PAMI-5, No.1, pp.149-163, 1983.

Derin, H. and H. Elliot, "Modeling and Segmentation of Noisy and Textured Images Using Gibbs Random Fields", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol.PAMI-9, No.1, pp.39-55, 1987.

Draper, B.A., R.T. Collins, J. Brolio, A.R. Hanson and E.M. Riseman, "The Schema System", *Intern. Journal of Computer Vision*, Vol.2, pp.209-250, 1989.

DuBuf, J.M.H., M. Kardan and M. Spann, "Texture Feature Performance for Image Segmentation", *Pattern Recognition*, Vol.23, No.3-4, pp.291-309, 1990.

Duda, O.R., Hart, P.E., Pattern Classification and Scene Analysis, John Wiley&Sons, 1973.

Fisher, D.H., "Knowledge Acquisition Via Incremental Conceptual Clustering", *Machine Learning*, Vol.2, pp.139-172, 1987.

Gennari, J.H., P. Langley and D. Fisher, "Models of Incremental Concept Formation", *Artificial Intelligence*, Vol.40, pp.11-61, 1989.

Goldfarb, L., "On the Foundations of Intelligent Processes - I. An evolving model for pattern learning", Pattern Recognition, Vol.23, No.6, pp.595-616, 1990.

Goto, Y. and A. Stentz, "Mobile Robot Navigation: The CMU System", *IEEE Expert*, pp.44-54, Winter, 1987.

Goto, Y, S. A. Shafer and A. Stentz, "The Driving Pipeline: A Driving Control Scheme for Mobile Robots", Report CMU-RI-TR-88-8, The Robotics Institute, Carnegie Mellon University, 1988.

Hsiao, J.Y. and A.A. Sawchuk, "Unsupervised Textured Image Segmentation Using Feature Smoothing and Probabilistic Relaxation Techniques", *Computer Vision, Graphics and Image Processing*, Vol.48, pp.1-21, 1989.

Jamshidi, M., Large-Scale Systems: Modeling and Control, North-Holland, 1983.

Kanatani, K.-I. and T.-C. Chou, "Shape from Texture: General Principle", *Artificial Intelligence*, Vol.38, pp.1-48, 1989.

Laws, K.I., "Textured Image Segmentation", PhD Thesis, Dept. of Electrical Engineering, University of Southern California, Los Angeles, 1980.

Liedtke, C.-E. and M. Ender, "A Knowledge Based Vision System for the Automated Adaptation to New Scene Contents", Proc. 8th Int. Conf. on Pattern Recognition, Paris, pp.795-797, 1986.

Liu, S-S., and M.E. Jernigan, "Texture Analysis and Discrimination in Additive Noise", *Computer Vision, Graphics and Image Processing*, Vol.49, pp.52-67, 1990.

Matsuyama, T., "Expert Systems for Image Processing : Knowledge-Based Composition of Image Processes", *Computer Vision, Graphics and Image Processing*, Vol.48, pp.22-49, 1989.

Michalski, R.S. and J.B. Larson, "Selection of most representative training examples and incremental generation of VL1 hypotheses: the underlining methodology and descriptions of programs ESEL and AQ11", Report 867, Department of Computer Science, University of Illinois, Urbana, 1978.

Michalski, R.S. and R.L. Chilausky, "Learning by Being Told and Learning form Examples: An experimental comparison of the two methods of knowledge acquisition in the context of developing an expert system for soybean disease diagnosis", *Policy and Information Systems*, Vol.4, pp.125-160, 1980.

Michalski, R. S., "A Theory and Methodology of Inductive Learning", in Machine Learning: An Artificial Intelligence Approach, TIOGA Publishing, Palo Alto, CA, pp 83-134, 1983.

Mitchell, T.M., Invited talk, 6th Int. Symposium on Methodologies for Intelligent Systems, Charlotte, N.C., 1991.

Neumann, B., "Towards Computer Aided Vision System Configuration", in Artificial Intelligence II: Methodology, Systems, Applications, Ph. Jorrand and V. Sgurev (eds), pp.385-393, Elsevier Pub., 1987.

Niemann, H., H. Bruenig, R. Salzbrunn and S. Schroeder, "A Knowledge-Based Vision System for Industrial Applications", *Machine Vision and Applications*, Vol.3, pp201-229, 1990.

Nii, H. P., "Blackboard Systems, Blackboard Application System, Blackboard System from a Knowledge Engineering Perspective", *The AI Magazine*, pp.82-106, August, 1986.

Nii, H. P., "Blackboard Systems: Ther Blackboard Model of Problem Solving and the Evolution of Blackboard Architectures", *The AI Magazine*, pp.38-52, August, 1986.

Nillson, N. N., Learning Machines, 1965.

Pachowicz, P.W., "Low-Level Numerical Characteristics and Inductive Learning Methodology in Texture Recognition", Proc. IEEE International Workshop on Tools for AI, Washington, D.C., pp.91-98, October 1989.

Pachowicz, P.W., "Integrating Low-Level Features Computation with Inductive Learning Techniques for Texture Recognition", *International Journal of Pattern Recognition and Artificial Intelligence*, Vol.4, No.2, pp.147-165, 1990b.

Pachowicz, P.W., "Learning Invariant Texture Characteristics in Dynamic Environments: A model evolution approach", Report MLI-2-91, Center for Artificial Intelligence, George Mason University, 1991.

Pachowicz, P.W. and J. Bala, "Texture Recognition through Machine Learning and Concept Optimization", Report MLI-6, George Mason University, Center for Artificial Intelligence, 1991 (submitted also to the IEEE Tr. on Pattern Analysis and Machine Intelligence).

Padulo, L. and M.A. Arbib, "System Theory", W.B. Saunders Company, 1974.

Reinke, R.E. and R.S. Michalski, "Incremental Learning of Concept Descriptions: A Method and Experimental Results", Machine Intelligence 11, J.E. Hayes, D. Michie and J. Richards (Eds), Clarendon Press, Oxford, pp.263-288, 1988.

Rine, D., "Retrainable Software: Software Engineering and Machine Learning", Digest of the AIDA-88 Conference, George Mason University, November 1988.

Rine, D., "A Formal Approach to Perfective Maintenance: using a basis", to appear in the *Int. J. of Software Maintenance*, 1991.

Rine, D., "Software Maintenance: by means of retrainable software", to appear in the *Int. J. of Software Maintenance*, 1992.

Roan, S. J., J.K. Aggarwal and W.N. Martin, "Multiple Resolution Imagery and Texture Analysis", *Pattern Recognition*, vol.20, No.1, pp.17-31,1987.

Rosenfeld, A. and L. Davis, "Image Segmentation and Image Models", *Proc. of IEEE*, Vol.67, No.12, pp. 7646-7772, 1979.

Rosenfeld, A., J. Kender, M. Nagao, L. Uhr, W.B. Thompson, V.A. Kovalevsky, D. Sher and S. Tanimoto, "DIALOG Expert Vision Systems: Some Issues", *Computer Vision, Graphics and Image Processing*, Vol.34, pp.99-117, 1986.

Sastry, S. and Bodson, M., "Adaptive Control: Stability, Covergence and Robustness", Prentice Hall, 1989.

Spiessbach, A.J., "A Control Architecture for a Mars Walking Vehicle", *Proc. SPIE - Intelligent Control and Adaptive Systems*, Vol. 1196, pp.2-13, 1989.

Tomita, F. and S. Tsuji, "Extraction of Multiple Regions by Smoothing in Selected Neighborhoods", *IEEE Tr. on Systems, Man and Cybernetics*, Vol.SMC7, pp.107-109, 1977.

Unser, M. and M. Eden, "Multiresolution Feature Extraction and Selection for Texture Segmentation", *IEEE Trans. Pattern Analysis and Mach. Intel.*, Vol.PAMI-11, No.7, pp.717-728, 1989.

Utgoff, P.E., "Incremental Induction of Decision Trees", *Machine Learning*, Vol.4, pp.161-186, 1989.

Weszka, J.S., C.R. Dyer and A. Rosenfeld, "A Comparative Study of Texture Measures for Terrain Classification", *IEEE Trans. Systems Man Cybern.*, Vol.SMC-6, No.4, pp.269-285, 1976.

Whitehall, B.L., S.C-Y. Lu and R.E. Step, "CAQ: A Machine Learning Tool for Engineering", *International Journal for Artificial Intelligence in Engineering*, Vol.5, No.4, 1990.